

UNCLASSIFIED

AD 288 438

*Reproduced
by the*

ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

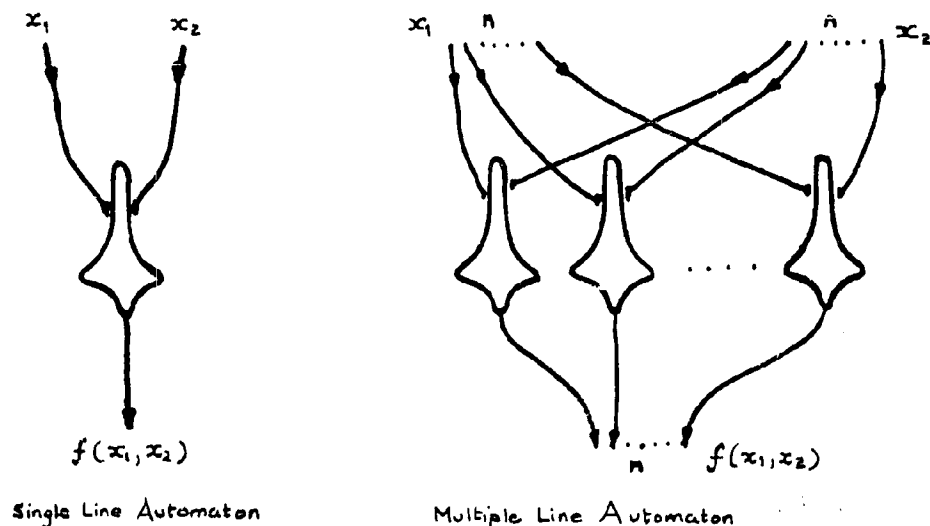
TOWARD A PROPER LOGIC FOR PARALLEL COMPUTATION IN THE PRESENCE OF NOISE

Jack D. Cowan*

Massachusetts Institute of Technology
Research Laboratory of Electronics

INTRODUCTION

The work of J. von Neumann,¹ and of W. S. McCulloch,² M. Blum,³ and L. A. M. Verbeek,⁴ on the construction of reliable automata from unreliable components has demonstrated the effectiveness of introducing redundancy into the structure of such automata, by means of the "bundling" technique. That is, replacing each single-line automaton by a parallel net of single-line automata, subject to the constraint that "bundles" in the multiple-line automaton, carry the same amount of information as do individual lines in the single-line automaton. Thus if there are n lines per bundle, each line carries $1/n^{\text{th}}$ of a bit, compared with 1 bit per line for the single-line automaton. The number of lines per bundle is a measure of the redundancy of parallel nets. The informational constraint is introduced as follows:



Single-line automata are replaced by multiple-line automata (Figure 1) that possess n lines per bundle. A fiduciary level Δ is set such that

* Also Visiting Research Associate, Department of Electrical Engineering, Northeastern University, Boston, Massachusetts, and Communication Theory Group, U. S. Air Force Cambridge Research Laboratory, Bedford, Massachusetts.

*
CATALOGED BY ASTIA
AS AD NO. 288438

If more than $(1-\Delta)n$ lines "fire", this is interpreted as "1".
 If less than Δn lines "fire", this is interpreted as "0".
 Any intermediate firing level is interpreted as a malfunction,
 "i".

In the construction used by McCulloch et al., the same coding procedure is followed, but the multiple-line automaton has the following structure:

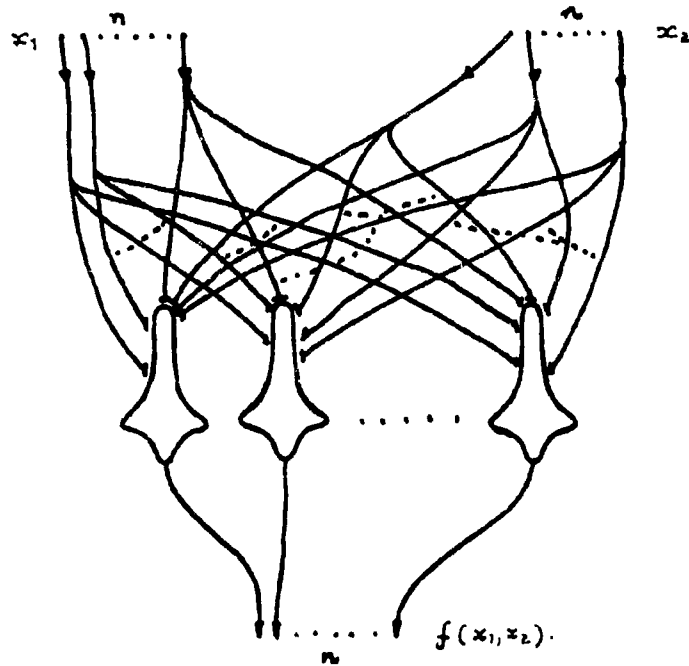


Figure 2

Thus each "neuron" receives inputs from all lines in both bundles. We shall not enter into an analysis of these particular systems (see Löfgren⁵ and Cowan⁶) but merely state the results. For both constructions, if ϵ is the probability of malfunction associated with each component, and P_e is the probability of malfunction of a complete net, we obtain the following relation:

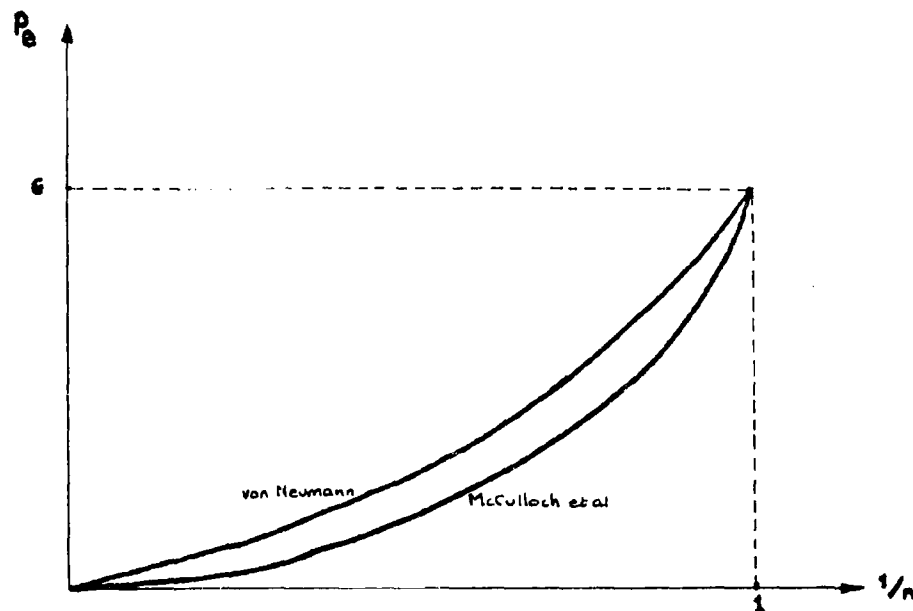


Figure 3

Thus P_e is a monotonically increasing function of $1/n$, and in both cases $\lim_{n \rightarrow \infty} P_e = 0$. But $1/n$ is the number of bits per "line", and is thus a measure of the information rate (see Shannon⁷) in the multiple-line automaton. So these results both have the property that the rate is zero for reliable computation in the presence of noise. When we consider Information Theory,⁷ this result is very surprising. The noisy coding theorem may be stated as follows:

If, for a communication system, there exists a certain maximum rate for transmission of information, the channel capacity C , then for transmission rates R less than C , it is possible to introduce redundancy, independent of rate, in such a way as to obtain an arbitrarily small error probability, P_e . At rates higher than C , P_e increases with $R - C$.

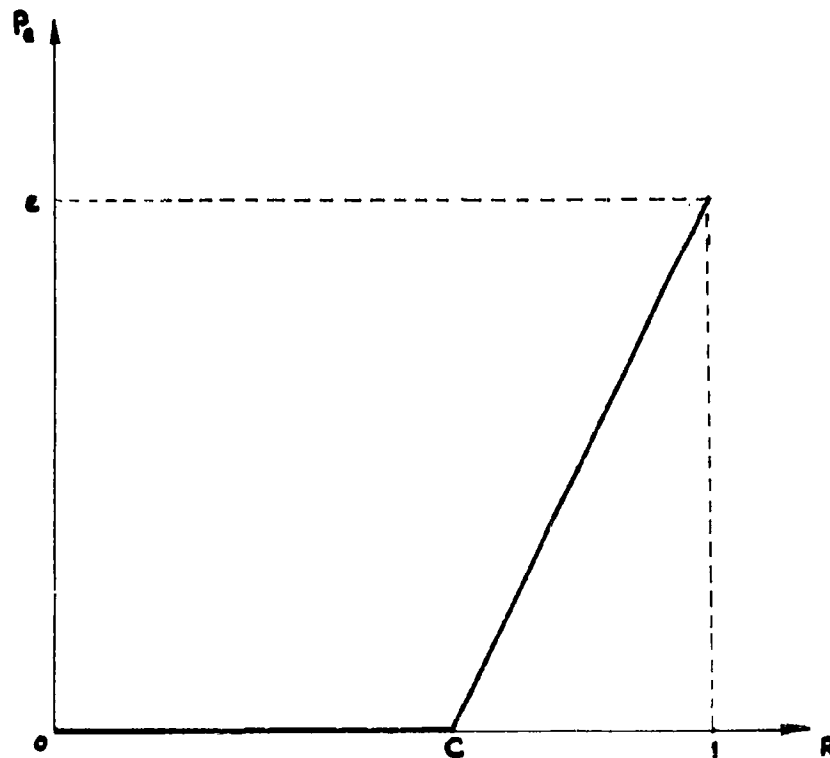


Figure 4

The results of von Neumann and McCulloch et al. would seem to imply that the capacity for computation of information is zero. These results motivated Elias⁸ to attempt an information-theoretic solution to the problem. He noted that only the odd-jot functions of 2-variable, 2-valued logic presented any problem (see Blum³). For these functions a block code (k information digits, $n - k$ check digits) allows correction of all errors as the block length n increases indefinitely, only when the rate k/n is, at most, $1/n$. (In fact, the rate can never be greater than $1/3$ for correction of single errors.) Thus block coding is no better than the iterative coding of von Neumann and McCulloch et al. On the basis of this result, Elias hypothesized that the computational capacity is zero. Petersen⁹ carried through a similar argument for binary group codes and achieved essentially the same results.

We shall now attempt to define a particular information measure relevant to computation, as distinct from transmission, in order to gain insight into the nature of the foregoing results, and attempt a solution to the problem of realizing a nonzero rate for reliable computation.

COMPUTATIONAL CAPACITY

A computing device differs from a transmitting device, in that, in general, the former is "dimension-reducing," while the latter is "dimension-preserving." Thus we can characterize computing devices as follows:

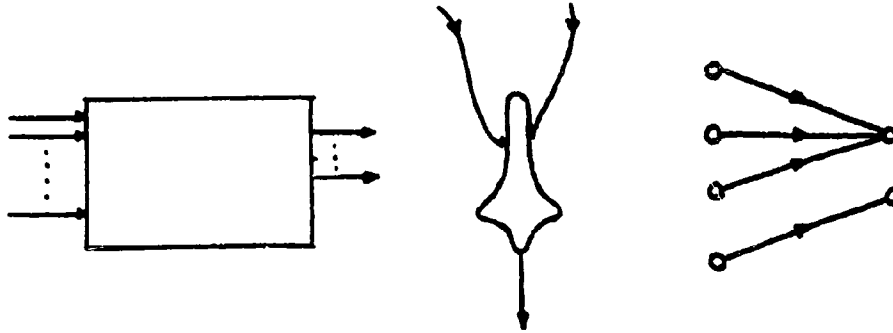


Figure 5

and transmitting devices as follows:

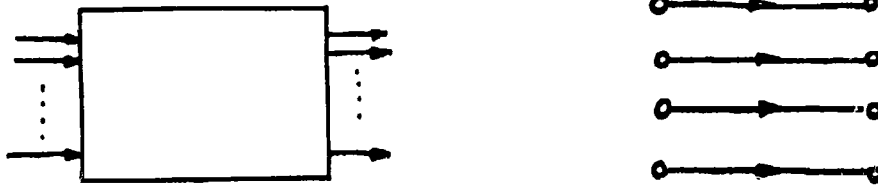


Figure 6

The two systems can be distinguished as follows. Consider both as mappings from an input-space \underline{X} into an output-space \underline{Y} . Then, in the absence of noise,

(2.1) For transmission

$$H(\underline{X}|\underline{Y}) =_{df} \sum_{\underline{X}} \sum_{\underline{Y}} P_{\{x,y\}} \log P_{\{x|y\}} = 0$$

(2.2) For computation,

$$H(\underline{X}|\underline{Y}) > 0$$

The quantity $H(\underline{X}|\underline{Y})$, called the communication entropy of \underline{X} given \underline{Y} , is called equivocation.

We can also define the mutual information rate for $\underline{X} \rightarrow \underline{Y}$:

$$\begin{aligned} \mathcal{I}[\underline{X}; \underline{Y}] &=_{df} \sum_{\underline{X}} \sum_{\underline{Y}} P_{\{x,y\}} \log \frac{P_{\{x,y\}}}{P_{\{x\}} P_{\{y\}}} \\ &= H(\underline{X}) - H(\underline{X}|\underline{Y}) \\ &= H(\underline{Y}) - H(\underline{Y}|\underline{X}) \end{aligned}$$

Clearly, $H(\underline{X}|\underline{Y}) = H(\underline{Y}|\underline{X})$ implies $H(\underline{X}) = H(\underline{Y})$.

Since $H(\mathcal{K}) =_{df} - \sum_{\mathcal{K}} P_{\{u\}} \log P_{\{u\}}$ is in a certain sense a measure of $\dim \mathcal{K}$, the dimension of the space \mathcal{K} (see Rényi¹⁰), we require for computation to occur in $\underline{X} \rightarrow \underline{Y}$ that $H(\underline{Y}) < H(\underline{X})$, and hence $H(\underline{Y}|\underline{X}) < H(\underline{X}|\underline{Y})$. That is, the noisy entropy must be less than the equivocation. In general, a computing scheme exists when

$$(2.3) \quad H(\underline{X}|\underline{Y}) > 0 \quad \text{in the absence of noise.}$$

$$(2.4) \quad H(\underline{Y}|\underline{X}) < H(\underline{X}|\underline{Y})$$

Consider, now, the following scheme:

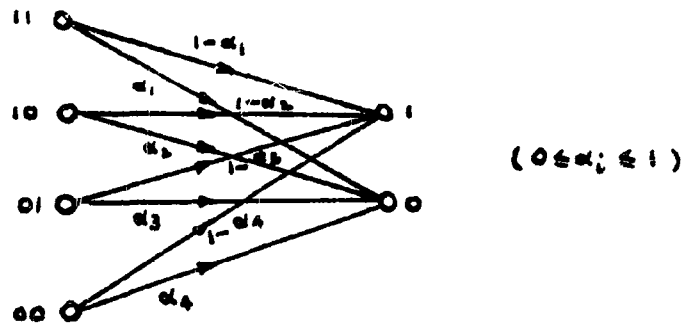


Figure 7

or equivalently McCulloch's chiastic scheme.¹¹

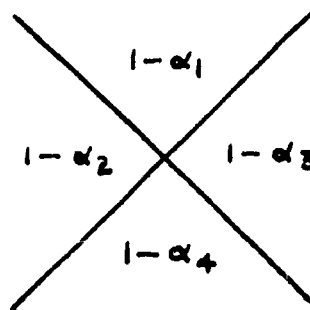


Figure 8

This represents a computing scheme in which, not the argument, but the function, is probable. By specification of the a_i we represent all possible Boolean logical functions, and their perturbations by noise or malfunction. For example, suppose $(a_1, a_2, a_3, a_4) = (\epsilon, \epsilon, \epsilon, 1-\epsilon)$. Then we have specified the following scheme:

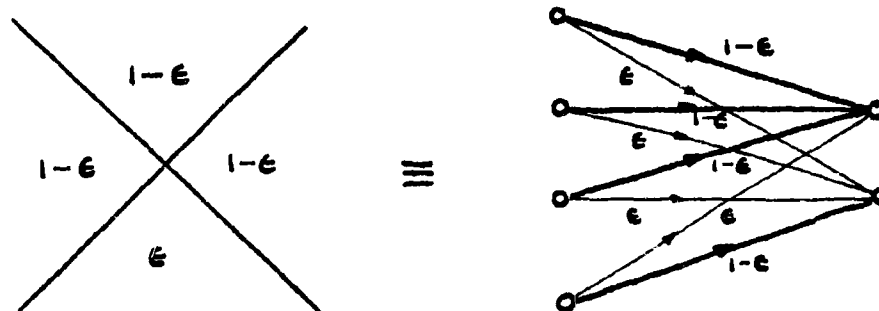


Figure 9

We interpret this as the function " x_1 or x_2 "

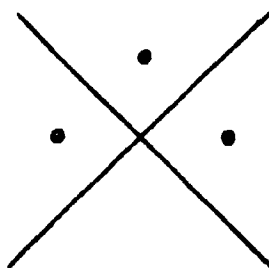


Figure 10

together with the fact that with each possible input-output pair, there is associated a precise probability of error ϵ .¹

We can formally represent this, by attaching a binary symmetric channel (BSC) to the output of the computing device:

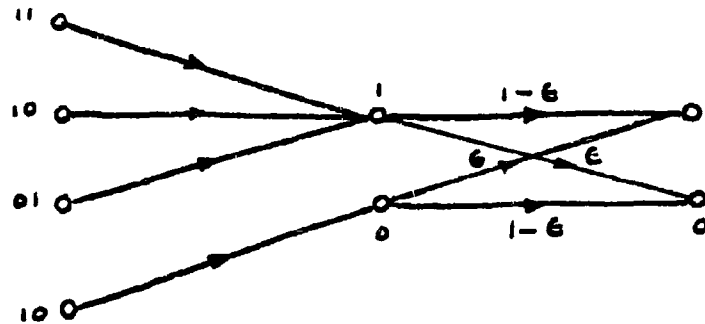


Figure 11

Of course, we need not attach a BSC, capacity $C_{BSC} = 1 - H(\epsilon)$
 $= 1 + \epsilon \log_2 \epsilon + (1-\epsilon) \log_2 (1-\epsilon)$ to the output. We could have
 used any noisy binary channel, with capacity C at the output. For sim-
 plicity, we shall consider here only the former case. Furthermore, we
 shall consider only systems with two inputs (x_1, x_2). In general, of
 course, we can have any number of inputs.^a

^a

The general $n \rightarrow 1$ computational scheme (n finite), with any noisy binary channel at the output is being considered by S. Winograd, Research Laboratory of Electronics, M. I. T., and the writer.

We are thus considering the following scheme:

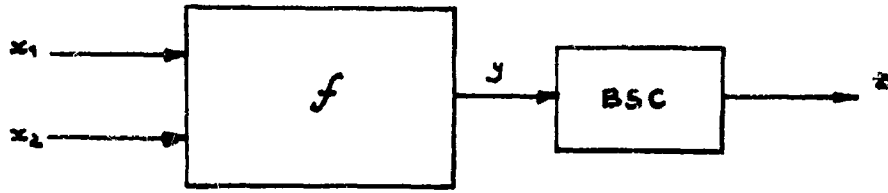


Figure 12

which we symbolize as $X_1 \otimes X_2 \xrightarrow{f} Y \xrightarrow{g} Z$. Since f is noiseless, the capacity for $X_1 \otimes X_2 \xrightarrow{f} Y$ is just 1 bit, while the capacity for $Y \xrightarrow{g} Z$ is C_{BSC} . And since the two channels are in series, the maximum rate for $X_1 \otimes X_2 \xrightarrow{f \circ g} Z$ is just $\min(1, C_{BSC}) = C_{BSC}$. That is, the maximum mutual information rate that can be obtained from the noisy computation scheme $X_1 \otimes X_2 \xrightarrow{f \circ g} Z$ is just the capacity of the dimension-preserving channel specifying the noise in the scheme.

We can compute this explicitly as follows: we note that

$$(2.5) \quad 0 \leq H(X_1 \otimes X_2) \leq 2$$

$$(2.6) \quad 0 \leq H(Y) \leq 1$$

$$(2.7) \quad H(Y | X_1 \otimes X_2) = 0$$

$$(2.8) \quad 0 \leq H(X_1 \otimes X_2 | Y) \leq H(X_1 \otimes X_2)$$

$$(2.9) \quad \begin{aligned} H(X_1 \otimes X_2) - H(X_1 \otimes X_2 | Y) \\ = H(Y) - H(Y | X_1 \otimes X_2) \end{aligned}$$

From (2.7) and (2.9)

$$H(Y) = H(X_1 \otimes X_2) - H(X_1 \otimes X_2 | Y)$$

That is,

$$J[X_1 \otimes X_2; Y] = H(Y)$$

Clearly,

$$\begin{aligned} J[X_1 \otimes X_2; Z] &= J[X_1 \otimes X_2; Y] - H(Y|Z) \\ &= H(Y) - H(Y|Z) \end{aligned}$$

Thus the capacity for $X_1 \otimes X_2 \xrightarrow{\text{tag}} Z$ is just $J_{\max}[X_1 \otimes X_2; Z] = \max[H(Y) - H(Y|Z)] = C_{BSC}$. (q. e. d.)

From this simple analysis, it would appear that essentially the same results that apply to the noisy computation channel apply to the noisy transmission channel, namely: up to a certain mutual information rate, C_{BSC} , it is possible to introduce redundancy into the system, independent of rate, and achieve an arbitrarily small probability of error in the computation. The question is, of course, can we, in fact, introduce redundancy into a computing scheme, so as to realize this nonzero rate for reliable computation, and if so, what is the nature of this redundancy? Before trying to answer this question, we shall analyze the nature of the measure $J[X_1 \otimes X_2; Z]$.

THE NATURE OF $J[X_1 \otimes X_2; Z]$

For transmission, the information measure that we have used is $J[X; Y]$, which we interpret as a measure of the average rate at which information about "points" in X is provided by reception of "points" in Y . In the case of computation, the measure that we have used is $J[X_1 \otimes X_2; Z]$, where

$$\begin{aligned} J[X_1 \otimes X_2; Z] &= H(X_1 \otimes X_2) - H(X_1 \otimes X_2 | Z) \\ &= H(Z) - H(Z | X_1 \otimes X_2) \end{aligned}$$

Furthermore,

$$\begin{aligned} \mathcal{I}[X_1 \otimes X_2; \bar{Z}] &= H(\bar{Y}) - H(\bar{Y} | \bar{Z}) \\ &= H(\bar{Z}) - H(\bar{Z} | \bar{Y}) \end{aligned}$$

where

$$\bar{Y} = f \circ \bar{X}_1 \otimes \bar{X}_2$$

Thus

$$H(\bar{Z} | X_1 \otimes X_2) = H(\bar{Z} | f \circ X_1 \otimes X_2)$$

This is a consequence of the fact that $H(f \circ X_1 \otimes X_2 | X_1 \otimes X_2) = 0$ since f is noiseless. Thus

$$\begin{aligned} \mathcal{I}[X_1 \otimes X_2; \bar{Z}] &= H(\bar{Z}) - H(\bar{Z} | X_1 \otimes X_2) \\ &= H(\bar{Z}) - H(\bar{Z} | f \circ X_1 \otimes X_2) \\ &= \mathcal{I}[f \circ X_1 \otimes X_2; \bar{Z}] \\ &=_{df} \mathcal{I}[f; f^*] \end{aligned}$$

where

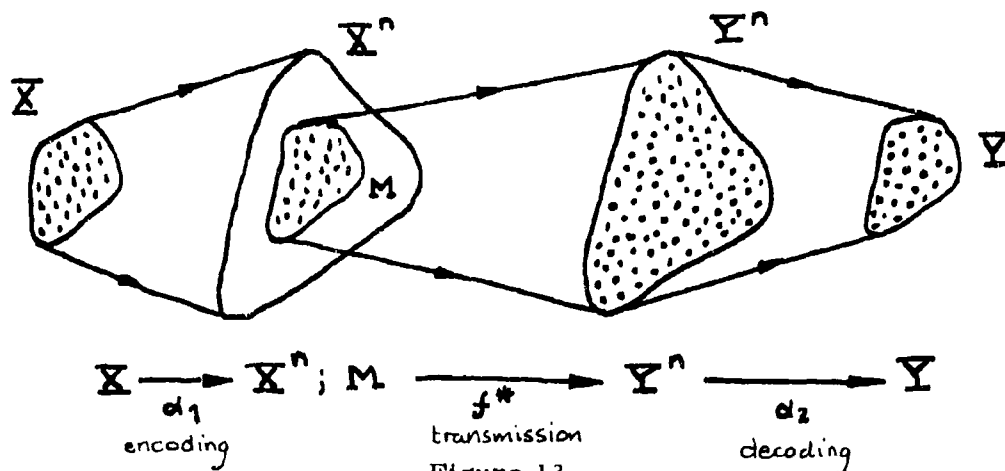
$$f^* =_{df} f \circ g.$$

This computation measure is thus a measure of the average rate at which information about points in $f \circ X_1 \otimes X_2$ is provided by reception of "points" in $f^* \circ X_1 \otimes X_2$. But clearly, points in $f \circ X_1 \otimes X_2$ correspond to point sets in $X_1 \otimes X_2$. That is $\mathcal{I}[f; f^*]$

is a measure of the average rate at which information about point sets in $X_1 \otimes X_2$, specified by f , is provided by reception of points in \bar{Z} . Since we are dealing with functions that are only probable,^b this is reasonable. Given a $z \in \bar{Z}$, we wish to compute, not what was the most likely input sequence $(x_1, x_2) \in X_1 \otimes X_2$, but what was the most likely input sequence subset, or equivalently, what was the most likely $f(x_1, x_2)$. Furthermore, since this is all that we wish to compute, no matter where malfunctions are located, and, for example, in formal neurons they can occur as threshold fluctuations, synapse errors, axon errors,⁴ as far as $\mathcal{J}[f; f^*]$ is concerned, these malfunctions are all effective at the output of f .

CODING FOR THE COMPUTATION CHANNEL

We now return to the question of coding for the computation channel, and our approach is based on certain implications of the noisy coding theorem for transmission channels. The whole process of introducing redundancy, encoding, decoding, and so forth, can be represented by the following set-theoretic picture:



^bWe note in this context, that the work of Shannon and Moore,¹² Kothen,¹³ and Allanson,¹⁴ is concerned not with probabilistic functions, but with functions of probabilistic arguments, and is different in character from the schemes considered here.

The theorem tells us that provided $\log M / \log X^n = \log M / n \log X$ is less than C_{BSC} , we can decode with arbitrarily small error probability P_e for the mapping $X \xrightarrow{\alpha_1, f \circ \alpha_2} Y$; that is, $\lim_{n \rightarrow \infty} P_e = 0$, provided $\log M / n \log X < C_{BSC}$. Thus, for reliable transmission $M < 2^{n \log X \cdot C_{BSC}}$, or, in general, $M = 2^{n \log X \cdot R}$, where $R = \mathcal{H}[X; Y]$.

For computation, $R = \mathcal{H}[f; f^*]$, and by analogy with transmission, we might expect to have $M = 2^{n \log f \circ X_1 \otimes X_2 \cdot \mathcal{H}[f; f^*]}$ "message" points in $f \circ X_1 \otimes X_2$: i. e.:

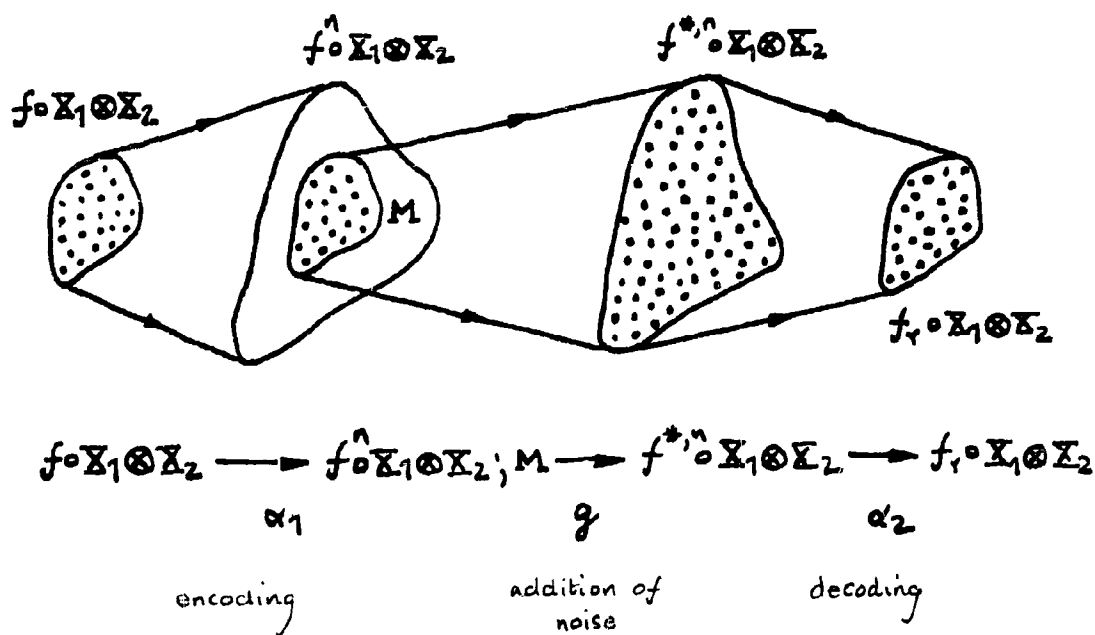


Figure 14

We distinguish between the function f_1 reconstructed by the decoding operation α_2 , and the initial function f . Thus $f_1 = (\alpha_1 \circ g \circ \alpha_2) \circ f$.

The function $f^* \circ X_1 \otimes X_2$ is interpreted to be a mapping of $X_1 \otimes X_2$ into a space containing a larger number of elements than the original function space $f \circ X_1 \otimes X_2$. The subspace (or subset) M is chosen as a "message" subset. Clearly, we are dealing with redundancy of function rather than redundancy of argument. Now for two-valued, two-variable logic, $f \circ X_1 \otimes X_2$ is a space with two elements, and hence, to obtain the necessary functional redundancy, we require a space $f^* \circ X_1 \otimes X_2$ with many elements, and a choice of only M of these to be informationally significant. That is, we have to construct a many-valued logical scheme in which not all truth values are informationally significant.⁶ This requirement is consistent with our location of the noise, and our consequent use of $g[f; f^*]$. From our equation $X_1 \otimes X_2 \xrightarrow{f \circ g} \bar{Z} = X_1 \otimes X_2 \xrightarrow{f} Y \xrightarrow{g} \bar{Z}$, it is evident that Y , not $X_1 \otimes X_2$, has to be matched to g , the noisy channel, and hence redundancy in Y is required. And Y is just the function space of f . It must be realized, however, that we are not free to perform $X_1 \otimes X_2 \xrightarrow{f} Y$, introduce redundancy in Y , and then perform $Y \xrightarrow{g} \bar{Z}$. On the contrary, we are really dealing with $X_1 \otimes X_2 \xrightarrow{f \circ g} \bar{Z}$. The other is merely a conceptual device, and the necessary redundancy may only be introduced into either $X_1 \otimes X_2$, or f , or both. At any rate, the coding scheme that we obtain, in general, is

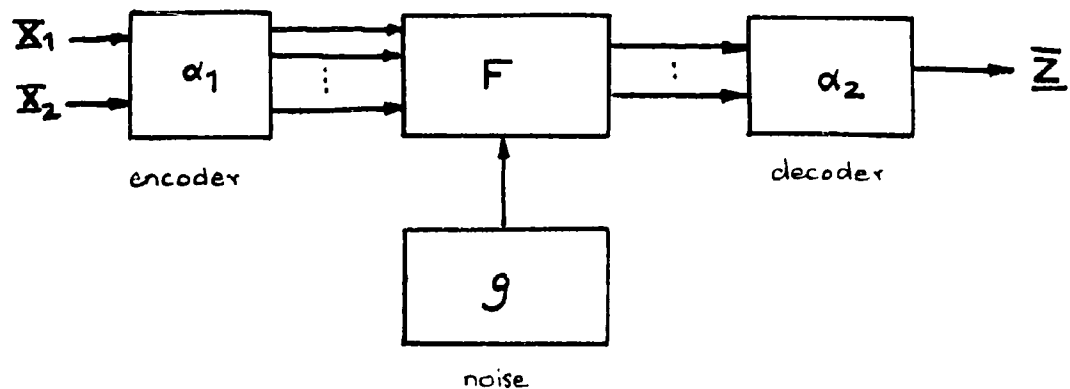


Figure 15

We write F for the ad hoc function, to indicate that it may be anything from f itself, to a much more complex function. We denote by M the encoded message set, that is $M = \alpha_1 \circ X_1 \oplus X_2$, and distinguish between $M = \underline{m} \oplus \underline{m}$, where $\underline{m} = \alpha_1 \circ X_1 = \alpha_1 \circ X_2$, which we call coordinate encoding, and $M \neq \underline{m} \oplus \underline{m}$, which we call pointwise encoding. We assume that α_1 , and α_2 are noiseless, and that no computation results from α_1 .⁸ The necessary conditions to be satisfied by the scheme can then be stated as follows: In the absence of noise, $\alpha_1 \circ F \circ \alpha_2 = f$, and is $[2:1]$ for all $x_i \oplus x_j \xrightarrow{\alpha_1 \circ F \circ \alpha_2} z$. For coordinate encoding this can be restated as $M \circ F = \underline{m} \oplus \underline{m} \circ F = \underline{m}$, $\underline{m} \circ \alpha_2 = f$, and $M \circ F$ is $[2:1]$ for all $m_i \oplus m_j \xrightarrow{F} m_k$. Furthermore, in the presence of noise, $d(M)$, the minimum distance between points of M , satisfies the Hamming inequality, namely that $d(M) \geq 2r + 1$, for correction of all r -tuple errors.¹⁵

MANY-VALUED LOGICAL SCHEMES

Let us now examine various forms of many-valued logical schemes. We consider initially a Post logical disjunction x_1 and x_2 (see Appendix) which, for the 3-valued scheme can be represented by the following truth-value matrix

x_2	1	2	3
1	1	2	3
2	2	2	3
3	3	3	3

which we also recognize as a mapping from the lattice of $X_1 \oplus X_2 \rightarrow X$, where $X = \{1, 2, 3\}$:

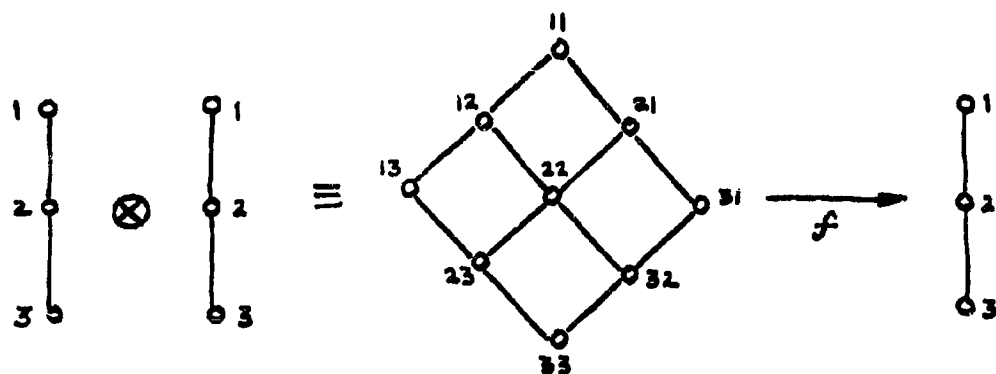


Figure 16

Let us choose $M = \{1, 3\}$, so that we have the following coordinate encoding scheme in the absence of noise:

\mathcal{C}	①	2	③
①	①	2	③
2	2	2	3
③	③	3	③

This scheme satisfies the conditions for coding, for single-error detection.^c Thus $d(M) = 2$, and the rate of computation is

^cNote that choice of $\{1, 3\}$ in this case preserves separation between code points. Thus if $d(1, 3) =$ distance between 1 and 3, then $d[f(1, 3), f(1, 3)] = [d(1, 3)]$.

$R = \log_2 N(M) / \log_2 m = \log_2 2 / \log_2 3$ b/s where $N(M)$ = number of elements in M , and m = number of truth-values in the logical scheme. We note that the product of rate and minimum distance is given by $Rd(M) = 2 / \log_2 3$. With $m = 5$, and $M = \{1, 3, 5\}$, we obtain the following mapping:

$\&$	① 2 ③ 4 ⑤
①	① 2 ③ 4 ⑤
2	2 2 3 4 5
③	③ 3 ③ 4 ⑤
4	4 4 4 4 5
⑤	⑤ 5 ⑤ 5 ⑤

This allows us to detect single errors at a rate $R = \log_2 3 / \log_2 5$ bits/symbol, $d(M) = 2$, and $Rd(M) = 2 \log_2 3 / \log_2 5$. Choosing $M = \{1, 5\}$ allows for single-error correction^d at a rate $R = \log_2 2 / \log_2 5$, with $d(M) = 4$, hence $Rd(M) = 4 / \log_2 5$. Similarly, with $m = 9$, $M = \{1, 4, 6, 9\}$, we obtain the following scheme:

$\&$	① 2 3 ④ 5 ⑥ 7 8 ⑨
①	① 2 3 ④ 5 ⑥ 7 8 ⑨
2	2 2 3 4 5 6 7 8 9
3	3 3 3 4 5 6 7 8 9
④	④ 4 4 ④ 5 ⑥ 7 8 ⑨
5	5 5 5 5 5 6 7 8 9
⑥	⑥ 6 6 ⑥ 6 ⑥ 7 8 ⑨
7	7 7 7 7 7 7 7 8 9
8	8 8 8 8 8 8 8 8 9
⑨	④ 9 9 ⑨ 9 ④ 9 9 ⑨

^dBy performing maximum likelihood computations, on the basis of minimum Hamming distance between "received" points and code points.¹⁶

which permits correction of some single errors, and detection of all others, at the rate $R = \log_2 4 / \log_2 9 = 1 / \log_2 3$, with $Rd(M) = 2 / \log_2 3$. This compares favorably with the 3-valued scheme, which permitted single-error detection only at the rate $1 / \log_2 3$. Alternatively, the 9-valued scheme permits single-error detection at this rate, with a lower probability of error P_e .

For, while $d(M)$ is constant, the mean distance between code points in M has clearly increased. With a 27-valued Post disjunction, and with $M = \{1, 5, 9, 12, 16, 19, 23, 27\}$, single errors can be detected at the rate $1 / \log_2 3$, but now $d(M) = 3$, and the mean code point separation has further increased, consequently P_e has decreased.

In general, for the M -valued Post disjunction, the mean separation between code points (that is, the mean nearest-neighbour separation) is

$$\delta(M) = m - 1 / m^R - 1. \text{ Since } R < 1, \text{ in the noisy case,}$$

$$\lim_{m \rightarrow \infty} \delta(M) = \infty. \text{ In particular, for } m = 3^n, R = 1 / \log_2 3,$$

$\delta(M) = 3^n - 1 / 2^n - 1, \lim_{m \rightarrow \infty} \delta(M) = \infty$. Thus, the mean separation between code points increases with m (or n), independently of the rate. Since the probability of error P_e associated with the coding scheme is inversely proportional to $\delta(M)$, this result would seem to imply that P_e approaches zero, independently of rate. This is not, however, the case, as examination of Post logic clearly demonstrates. The structures that we have considered are all mappings from direct products of chains, onto chains (see Appendix):

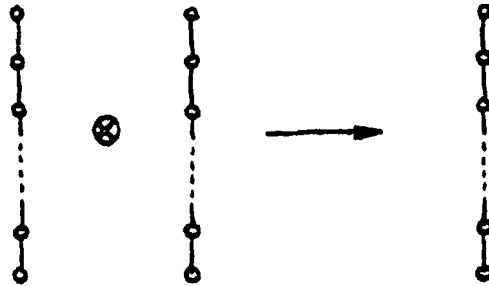


Figure 17

Increasing m , in such a way that this chain structure is preserved,

implies a transition from systems with, say, 3 states, ordered as follows:

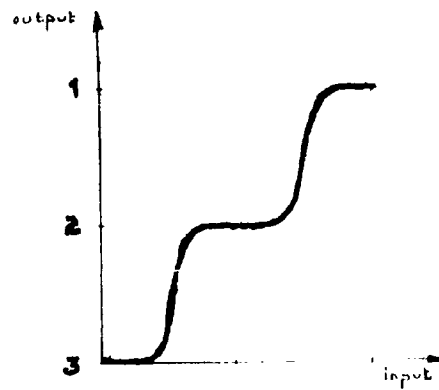


Figure 18

to systems with m states, ordered as follows:

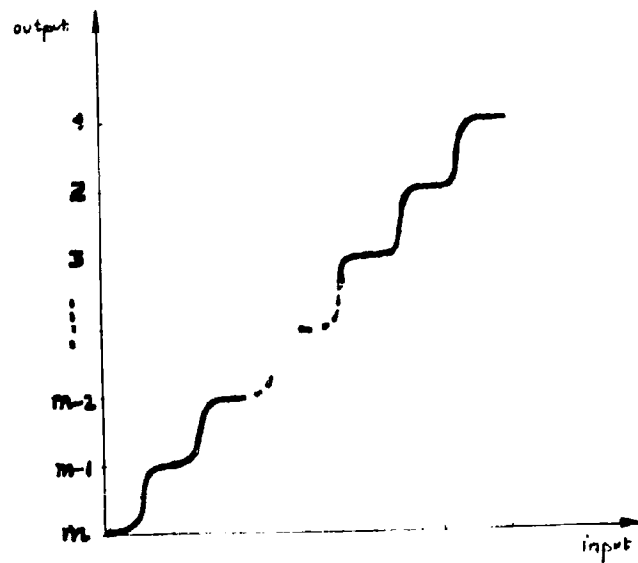


Figure 19

In the limit, we approach systems with a continuum of states. In terms of our model for computation, we have replaced the binary scheme (Figure 12) by the following m -ary scheme:

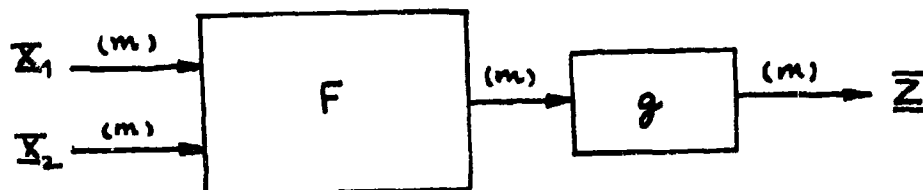


Figure 20

This means that we have replaced a binary channel (BSC), by a noisy m -ary channel:

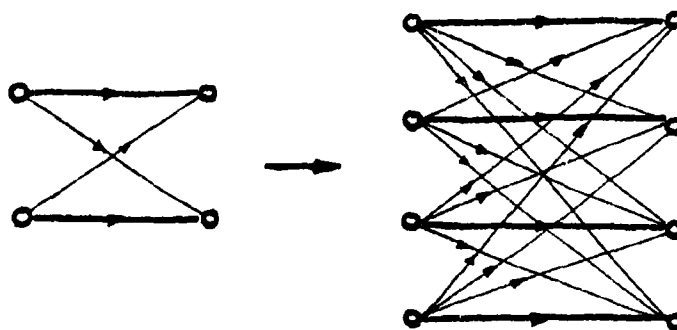


Figure 21

We can only gain by this, if the relative capacity of this channel is not less than the relative capacity of the binary channel. This will be the case (roughly) if the transition probabilities between nearest states of the m-ary channel are of the same order as those of the binary channel. Thus effectively, the m-ary scheme is

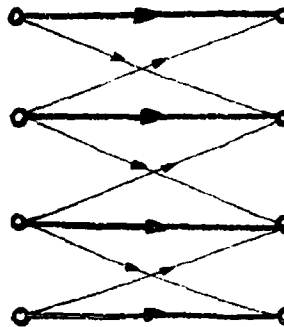


Figure 22

and all other transition probabilities are small enough to neglect. This condition is realizable with m-ary state schemes, if the noise fluctuations between neighbouring states are independent of M. In general, this is not true as M increases indefinitely, since components and transmission lines are energy-bounded, and we are limited to a fixed range in which to place states. As the number of states increases, the fluctuations between states become larger, and so the m-ary state channel of our model becomes effectively noisier. We can compute this effect, roughly, by normalizing $S(M)$. We note that if a_3 is the distance between states in the 3-valued scheme (so that the total range is of length $2a_3$), then the distance between neighbouring states in an m-ary scheme is $a_m = 2a_3 / m - 1$:

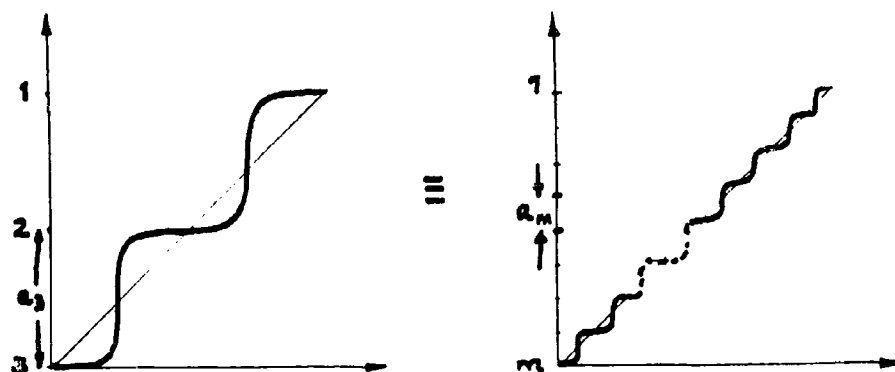


Figure 23

Then $\delta(M)$ normalized becomes $\delta^*(M) = \{2a^3(m/m-1)\} - 1/m^R - 1$.
 Choosing a_3 as our unit, we obtain $\delta^*(M) = \{2m/m-1\} - 1/m^R - 1 \sim (m^R)^{-1}$
 for large m , so that $\lim_{m \rightarrow \infty} \delta^*(M) = 0$. This implies that if we keep the rate constant, $\frac{m}{Pe}$ goes to 1, not to 0. This result is obvious, since $m \rightarrow \infty$ implies analog computation, and the notion of code redundancy is not meaningful. Evidently, we have to keep m fixed, and small, and investigate possibilities of obtaining Post functions with requisite minimum distance properties, by constructing redundant nets of m state components. We shall consider, first, those results obtained by the use of redundant parallel nets of 2 state components, that is, nets considered by von Neumann¹ and McCulloch et al.²⁻⁴ It has been shown,⁶ that the redundant automata constructed by von Neumann can be characterized by Lewis' many-valued logical schemes. For example, the following redundant "and" network:

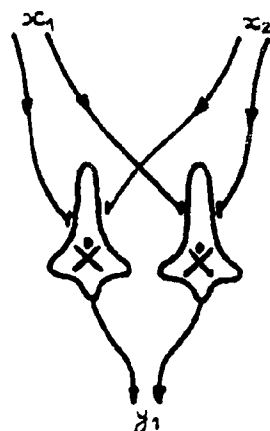


Figure 24

can be characterized by the 4-valued Lewis ' $\&$ ' :

$\&$	1	2	3	4
1	1	2	3	4
2	2	2	4	4
3	3	4	3	4
4	4	4	4	4

under the correspondence

$$\begin{pmatrix} 11 & 10 & 01 & 00 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

In lattice theoretic terms, this is equivalent to the mapping:

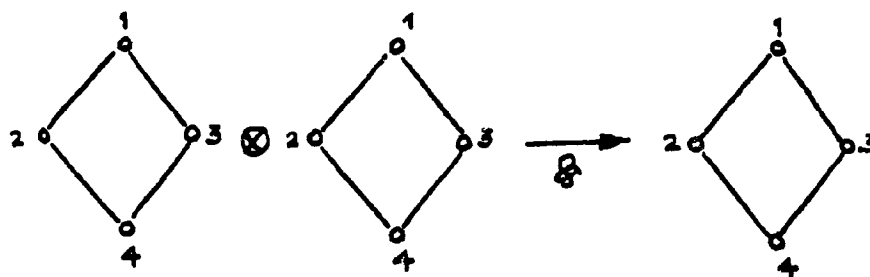


Figure 25

where the specificity of \mathfrak{g} is characterized by the following map:

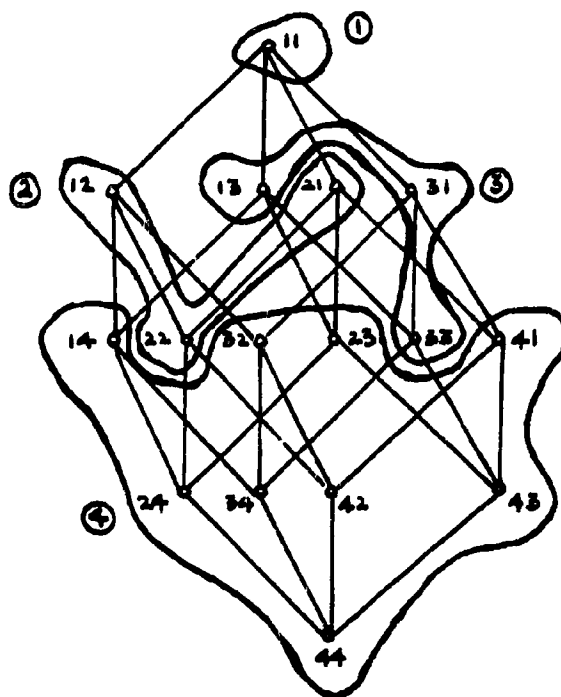


Figure 26

Suppose we choose $M = \{1, 4\}$. Then we obtain

\otimes	①	2	3	④
①	①	2	3	④
2	2	2	4	4
3	3	4	3	4
④	④	4	4	④

which satisfies the necessary coding requirements. We note that \otimes preserves minimum distance from $\mathbb{X}_1, \mathbb{X}_2$ to $\mathbb{X}_1 \otimes \mathbb{X}_2$. That is, the distance between subsets in $\mathbb{X}_1 \otimes \mathbb{X}_2$ ordered under \otimes^{-1} is equal to the minimum distance between distinct points in either \mathbb{X}_1 or \mathbb{X}_2 . For this scheme, we can evidently detect single errors at a rate $R = \log_2 2 / \log_2 4 = 1/2$, where $\delta(M) = 2$.

For $M = 2^3$ (3 lines per bundle),

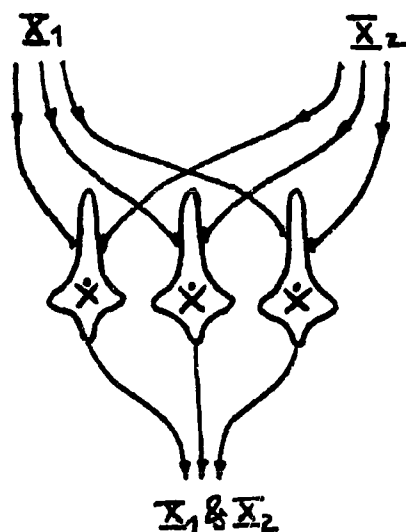


Figure 27

we obtain the 8-valued Lewis \mathcal{L}_8

\mathcal{L}_8	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	2	4	4	6	6	8	8
3	3	4	3	4	7	8	7	8
4	4	4	4	4	8	8	8	8
5	5	6	7	8	5	6	7	8
6	6	6	8	8	6	6	8	8
7	7	8	7	8	7	8	7	8
8	8	8	8	8	8	8	8	8

which is equivalently a mapping characterized by:

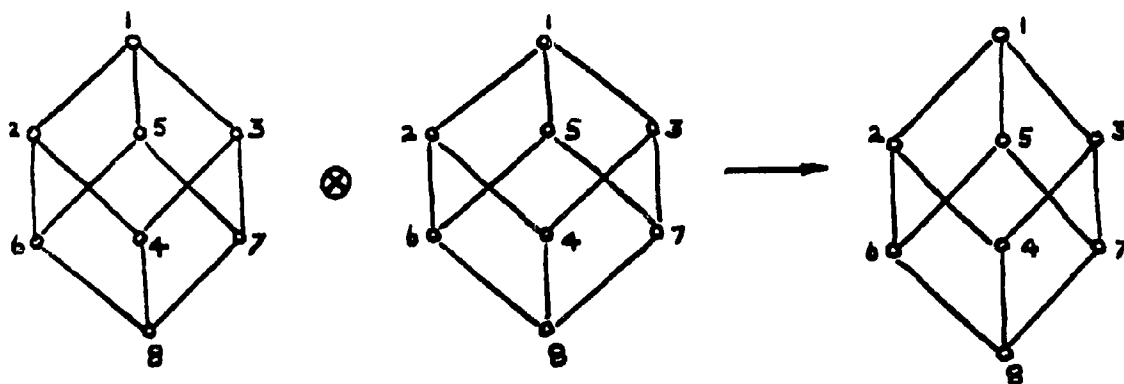


Figure 28

under the correspondence:

$$\begin{pmatrix} 111 & 110 & 101 & 100 & 011 & 010 & 001 & 000 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

When we come to choose our message set M , we find that the only set that satisfies our requirements is $M = \{1, 8\}$.

δ	①	2	3	4	5	6	7	⑧
①	①	2	3	4	5	6	7	⑧
2	2	2	4	4	6	6	8	8
3	3	4	3	4	7	8	7	8
4	4	4	4	4	8	8	8	8
5	5	6	7	8	5	6	7	8
6	6	6	8	8	6	6	8	8
7	7	8	7	8	7	8	7	8
⑧	⑧	8	8	8	8	8	8	⑧

We can thus correct single errors at the rate $R = \log 2 / \log 8 = 1/3$, with $\delta(M) = 3$.

Similarly, for $m = 2^4$, we obtain the 16-valued Lewis ' δ '. The only M -sets available are $M = \{1, 4, 13, 16\}$, and a subset of this, $M' = \{1, 16\}$. The former has $\delta(M) = 2$, so single errors are detectable at the rate $R = \log 4 / \log 16 = 1/2$, as before. The latter has $\delta(M) = 4$, so we can correct single errors and detect double errors at the rate $R = \log 2 / \log 16 = 1/4$. Alternatively, we can correct single errors at $R = 1/4$, $m = 2^4$, with lower P_e , than at $R = 1/3$, $m = 2^3$.

The nature of this result is clear. The coding conditions are always violated by any point set other than $\{1, 2^n\}$, or $\{1, 2^{n/2}, 2^n - 2^{n/2} + 1, 2^n\}$ (n even) or $\{1, 2^n\}$ (n odd), for the following reason: It should be evident that the truth-value lattice

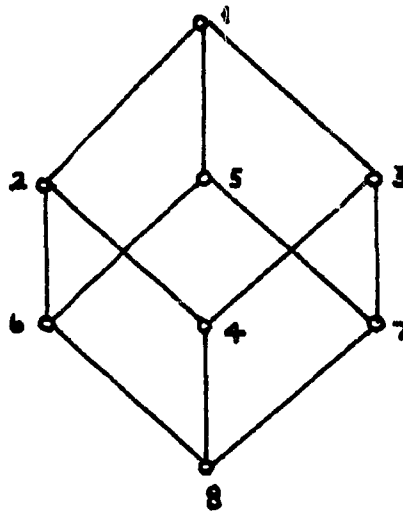


Figure 29

is itself a representation of the 8-valued Lewis ' \otimes ' (and its dual operation, the Lewis "or" function, ' \oplus '). Thus the infimum (g.l.b.) of any pair of lattice points is their ' \otimes '. For example, $2 \otimes 3 = 4$, $2 \otimes 7 = 8$, $5 \otimes 7 = 7$, $6 \otimes 7 = 8$, and so on. If we now choose any set other than $M = \{1, 8\}$, say $\{1, 4, 6, 7\}$, which has $\delta(M) = 2$, then closure is violated, since $4 \oplus 6 = 4 \otimes 7 = 6 \otimes 7 = 8$. This, in general, will always occur for odd-jot functions of 2 variable logic, since they possess the lattice property of always mapping any pair of points into a point an odd number of units of distance away. Thus the only point sets not violating the closure condition are those point sets containing only the points $1, 2^n$ and the extremal median lattice points whose intersection is the point 2^n . Thus the rate for error correction for these schemes is always, at most,

$$R = \log_2 4 / \log_2 2^n = 2/n, \quad n=6, 8, \dots$$

For the even-jot functions, these restrictions do not apply. Thus if we consider the following net: For $m = 2^3$ (3 lines per bundle),

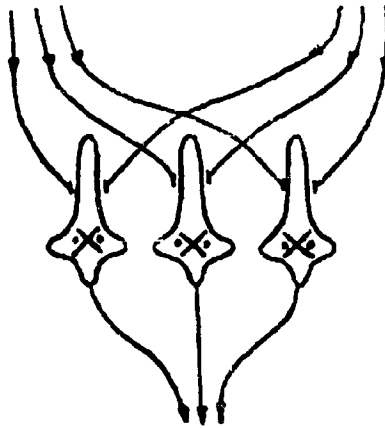


Figure 30

we obtain the 8-valued Lewis function

f^+	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	1	4	3	6	5	8	7
3	3	4	1	2	7	8	5	6
4	4	3	2	1	8	7	6	5
5	5	6	7	8	1	2	3	4
6	6	5	8	7	2	1	4	3
7	7	8	5	6	3	4	1	2
8	8	7	6	5	4	3	2	1

and if we now choose $M = \{1, 4, 6, 7\}$, the codability conditions are satisfied, and single errors can be detected at the rate $R = \log 4 / \log 8 = 2/3$, with $\delta(M) = 2$. Similarly, if we form the 16-valued Lewis function (4 lines per bundle), we obtain the function

f^+	①	2	3	④	5	⑥	⑦	8	9	⑩	⑪	12	⑬	14	15	⑯
①	①	2	3	④	5	⑥	⑦	8	9	⑩	⑪	12	⑬	14	15	⑯
2	2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15
3	3	4	1	2	7	8	5	6	11	12	9	10	15	16	13	14
④	④	3	2	①	8	⑦	⑥	5	12	⑪	⑩	9	⑯	15	14	⑬
5	5	6	7	8	1	2	3	4	13	14	15	16	9	10	11	12
⑥	⑥	5	8	⑦	2	①	④	3	14	⑬	⑯	15	⑩	9	12	⑪
⑦	⑦	8	5	⑥	3	④	①	2	15	⑯	⑬	14	⑪	12	9	⑩
8	8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9
9	9	10	11	12	13	14	15	16	1	2	3	4	5	6	7	8
⑩	⑩	9	12	⑪	14	⑬	⑯	15	2	①	④	3	⑥	5	8	⑦
⑪	⑪	12	9	⑩	13	⑯	⑬	14	3	④	①	2	⑦	8	5	⑥
12	12	11	10	9	16	15	14	13	4	3	2	1	8	7	6	5
⑬	⑬	14	15	⑯	9	⑩	⑪	12	5	⑥	⑦	8	①	2	3	④
14	14	13	16	15	10	9	12	11	6	5	8	7	2	1	4	3
15	15	16	13	14	11	12	9	10	7	8	5	6	3	4	1	2
⑯	⑯	15	14	⑬	12	⑪	⑩	9	8	⑦	⑥	5	④	3	2	①

If we now choose $M = \{1, 4, 6, 7, 10, 11, 13, 16\}$, single-error detection is possible at the rate $R = \log 8 / \log 16 = 3/4$, with $\delta(M) = 2$. In fact, in general, single-error detection in a 2^n valued Lewis logic can be performed for even jots at the rate $R = \frac{n-1}{n} = 1 - 1/n$. Similarly, single-error correction can be performed at a rate of, at most, $R = 1 - \frac{1}{n} \log_2(1+n)$, and so on. Alternatively, the rate can be kept constant, in which case the minimum distance increases, and hence P_e decreased.

In terms of the lattice picture, this result is possible because of the fact that Lewis functions composed of even-jot functions are such that any pair of points maps into a lattice point an even number of points away (apart from the set $\{1, 2\}$, which is always closed because Lewis logic is functionally incomplete). The particular coding used by von Neumann, "bundling" and setting a fiduciary level so that $R = 1/n$, can be characterized by the following diagram:

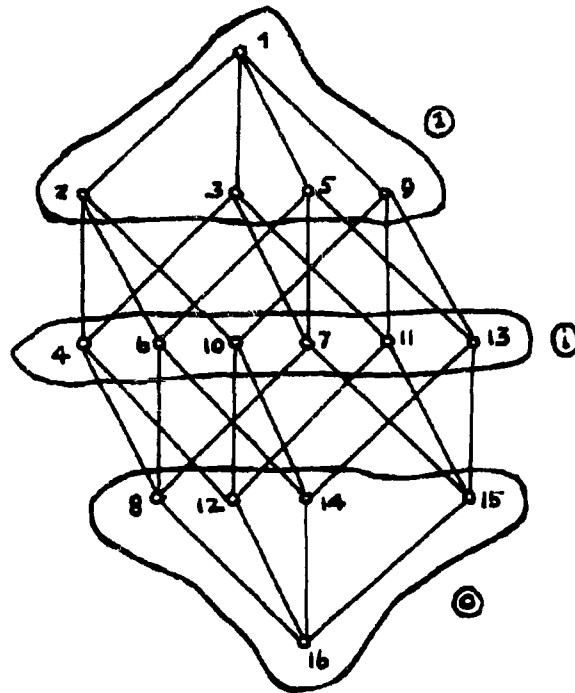


Figure 31

This is different in character from the previous schemes, in that there is no exclusion of possible inputs. The set M , as such, does not exist. What is being sought is a net which maps that which is designated as information, $\{1, 0\}$ into itself, and that which is designated as malfunction, $\{i\}$ into itself. For a fuller discussion of this, see Cowan.⁶

The scheme of McCulloch et al.²⁻⁴ uses essentially the same principle as this, but has a different structure. A more efficient error-reducing scheme results from the higher connectivity of the structure (see Cowan⁶).

Both schemes, however, are of the same character as the previous one, in that P_e goes to zero with R . All of these schemes are characterized by a certain symmetry — that the connectivity pattern for each neuron is identical for each scheme. We shall therefore investigate an asymmetric system

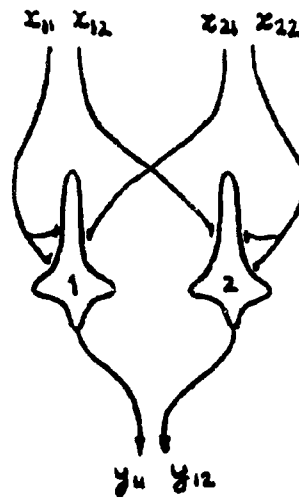


Figure 32

Let each neuron fire for inputs of strength 2, or more. Thus we have

		x_{21}		x_{22}	
		y_{11}		y_{12}	
x_{11}	2	1	1	1	0
	0	0	0	1	0

We combine these by taking their Kronecker product:

		\bar{X}_2				
		\bar{Y}	12	10	02	00
\bar{X}_1	21	11	10	11	10	
	20	11	10	11	10	
	01	10	00	10	00	
	00	10	00	10	00	

where

$$\bar{X}_1 = \{x_{11}, x_{12}\} ; \bar{X}_2 = \{x_{21}, x_{22}\} ; \bar{Y}_1 = \{y_{11}, y_{12}\} .$$

We now introduce the correspondences:

$$\bar{X}_1 : \begin{pmatrix} 21 & 20 & 01 & 00 \\ 1 & 2 & 3 & 4 \end{pmatrix} , \quad \bar{X}_2 : \begin{pmatrix} 12 & 10 & 02 & 00 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

and

$$\bar{Y}_1 : \begin{pmatrix} 11 & 10 & 01 & 00 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

and obtain the 4-valued Post function:

F	1	2	3	4
1	1	2	1	2
2	1	2	1	2
3	3	4	3	4
4	3	4	3	4

The only subsets not violating condition 1 are $\{1,2\}$ and $\{3,4\}$, neither of which satisfies condition 2.

Similarly, the following scheme (a modification of McCulloch's nets)

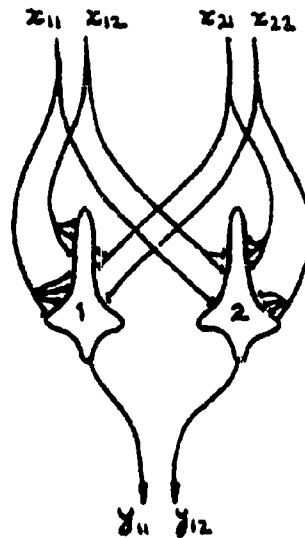


Figure 33

realizes the following Post function:

F	1	2	3	4
1	1	1	1	2
2	1	2	4	4
3	1	4	3	4
4	3	4	4	4

This, again, is an unsatisfactory mapping. It appears that any scheme with symmetric weights on the inputs, and noninteracting bundles will not work. However, more work needs to be done on this.

We now consider schemes with components possessing more than two stable states; and we consider, first, equal-input schemes, with $M = 3$.

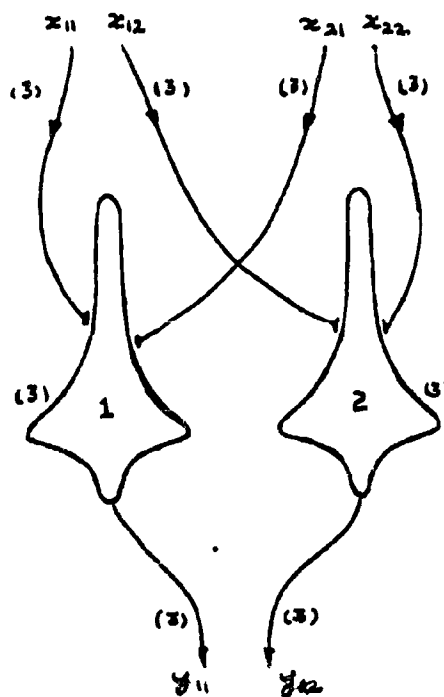


Figure 34

Each "neuron" has 3-stable states, and represents a Post ' \mathcal{L} ' :

\mathcal{L}	1	2	3
1	1	2	3
2	2	2	3
3	3	3	3

Taking the Kronecker product of this with itself, we obtain

$\mathcal{L} \otimes \mathcal{L}$	11	12	13	21	22	23	31	32	33
11	11	12	13	21	22	23	31	32	33
12	12	12	13	22	22	23	32	32	33
13	13	13	13	23	23	23	33	33	33
21	21	22	23	21	22	23	31	32	33
22	22	22	23	21	22	23	32	32	33
23	23	23	23	23	23	23	33	33	33
31	31	32	33	31	32	33	31	32	33
32	32	32	33	32	32	33	32	32	33
33	33	33	33	33	33	33	33	33	33

which leads to the 9-valued Lukasiewicz-Post function:

F	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	2	3	5	5	6	8	8	9
3	3	3	3	6	6	6	9	9	9
4	4	5	6	4	5	6	7	8	9
5	5	5	6	5	5	6	8	8	9
6	6	6	6	6	6	6	9	9	9
7	7	8	9	7	8	9	7	8	9
8	8	8	9	8	8	9	8	8	9
9	9	9	9	9	9	9	9	9	9

under the correspondence

$$\begin{pmatrix} 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{pmatrix}$$

Let us choose $M = \{1, 4, 6, 9\}$, so that $R = \log 4 / \log 9$
 ie, $R = 1 / \log_2 3$ We obtain the following scheme:

F	①	2	3	④	5	⑥	7	8	⑨
①	①	2	3	④	5	⑥	7	8	⑨
2	2	2	3	5	5	6	8	8	9
3	3	3	3	6	6	6	9	9	9
④	④	5	6	④	5	⑥	7	8	⑨
5	5	5	6	5	5	6	8	8	9
⑥	⑥	6	6	⑥	6	⑥	9	9	⑨
7	7	8	9	7	8	9	7	8	9
8	8	8	9	8	8	9	8	8	9
⑨	⑨	9	9	⑨	9	⑨	9	9	⑨

This is the same mapping (as far as informationally significant truth values are concerned) as $m=9, n=1$. The rate $R=1/\log_2 3$ is equal to that for $m=3, n=1$. However, the set $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, is not now a chain, but is itself a lattice. That is

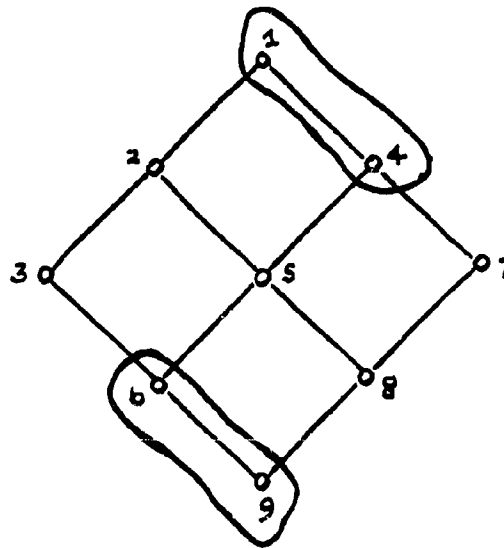


Figure 35

Since $d(1,4) = d(6,9) = 1$, condition 2 is violated, so that $\{1,4\}$ and $\{6,9\}$ are not separate information points. We might just as well send only $\{1,9\}$ at a rate $R = 1/2\log_2 3$. Let us, however, choose $M = \{1,3,7,9\}$. Then we obtain

F	①	2	③	4	5	6	⑦	8	⑨
①	①	2	③	4	5	6	⑦	8	⑨
2	2	2	3	5	5	6	8	8	9
③	③	3	③	6	6	6	⑨	9	⑨
4	4	5	6	4	5	6	7	8	9
5	5	5	6	5	5	6	8	8	9
6	6	6	6	6	6	6	9	9	9
⑦	⑦	8	⑨	7	8	9	⑦	8	⑨
8	8	8	9	8	8	9	8	8	9
⑨	⑨	9	⑨	9	9	9	⑨	9	⑨

We thus satisfy the coding conditions, so that single-error detection is possible, at the rate $R = 1 / \log_2 3$, with $d(M) = 2$; thus $R d(M) = 2 / \log_2 3$. Had we chosen $M = \{1, 9\}$, single-error correction would be possible at the rate $R = 1 / 2 \log_2 3$, with $d(M) = 4$, so that $R d(M) = 2 / \log_2 3$, as before. Similarly, choosing $m = 3^3 = 27$, and $M = \{1, 3, 7, 9, 19, 21, 25, 27\}$, we obtain the lattice

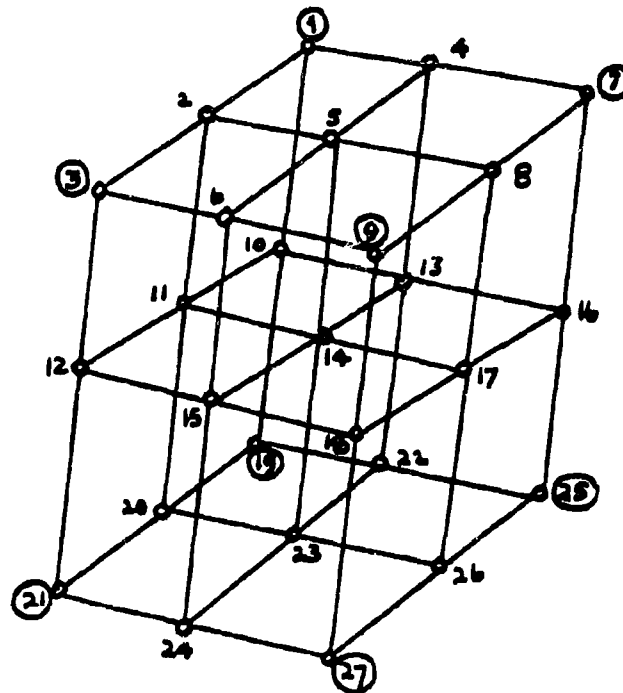


Figure 36

Single-error detection is possible, at the rate $R = 1 / \log_2 3$, $d(M) = 2$, and $Rd(M) = 2 / \log_2 3$, as before.

If we compare these schemes for $m = 3$, so that $Rd(M) = 2 / \log_2 3$ with those based on $m = 2$, wherein $Rd(M) = 1 / \log_2 2 = 1$, it would seem that a general relationship between m , R and $d(M)$ exists. In fact, for $m = 4$, $n = 2$, if we choose $M = \{1, 4, 13, 16\}$, we obtain the following scheme:

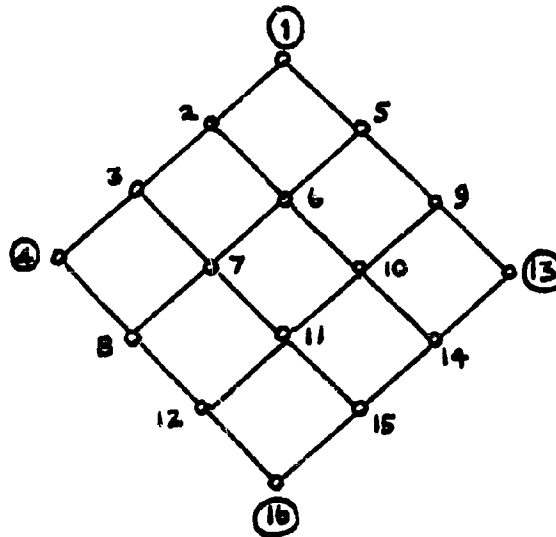


Figure 37

Thus $R = \log_2 4 / \log_2 16 = 1/2$, and $d(M) = 3$, so that $Rd(M) = 3/2 = 3 / \log_2 4$. In general, then, we can induce that $Rd(M) = (m-1) / \log_2 m$. (There are other reasons for the existence of this relation, based on notions of order continuity, and suchlike, which we shall not discuss here.) We have not, however, normalized $d(M)$, to account for limitations on the range available per component. If we perform this normalization, then $d(M)$ normalized is given by $d^*(M) = \{(m-r)/(m-1)\} d(M)$, $(1 \leq r \leq m-1)$. The parameter r measures the effects of different ratios of ranges for m states, to that of the 2 state range. Thus $r = 1$ implies that the distance between neighbouring states remains constant and the range is unbounded, while $r = m-1$ implies that the range is

fixed. These two limiting cases are of special interest. Thus

$$R_d^*(M) = \begin{cases} \frac{m-1}{\log_2 m}, & r=1 \\ \frac{1}{\log_2 m}, & r=m-1 \end{cases}$$

and therefore,

$$\lim_{m \rightarrow \infty} R_d^*(M) = \begin{cases} \infty, & r=1 \\ 0, & r=m-1 = \infty \end{cases}$$

we can plot these bounds as follows:

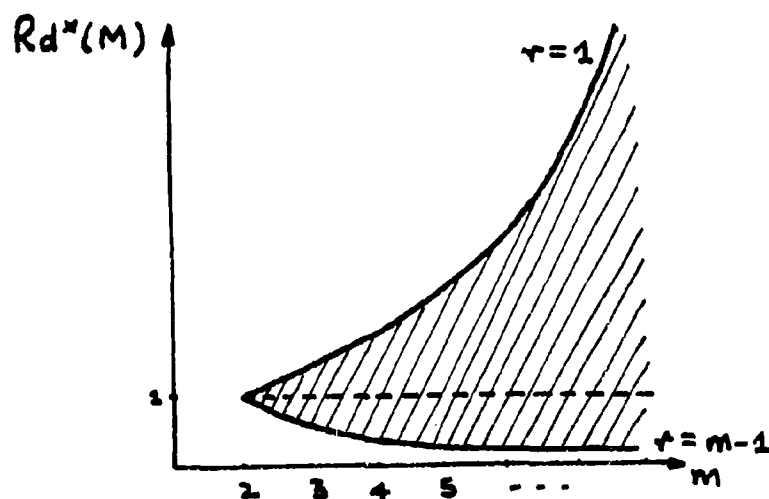


Figure 38

This figure can be interpreted as follows: Let $R_d^*(M) = \gamma(m)$. Then, since P_e varies inversely with $d(M)$, and since

$d(M) = \gamma(m)/R$, $P_e \sim R/\gamma(m)$. But, in general, for these product schemes, $R = 2/n \log_2 m$, hence $P_e \sim 2/n \log_2 m \cdot 1/\gamma(m)$. When $\tau = m-1$, $\gamma(m) = 1/\log_2 m$, and $P_e \sim 2/n$, which is independent of m . Hence we gain nothing by introducing m states. For example,

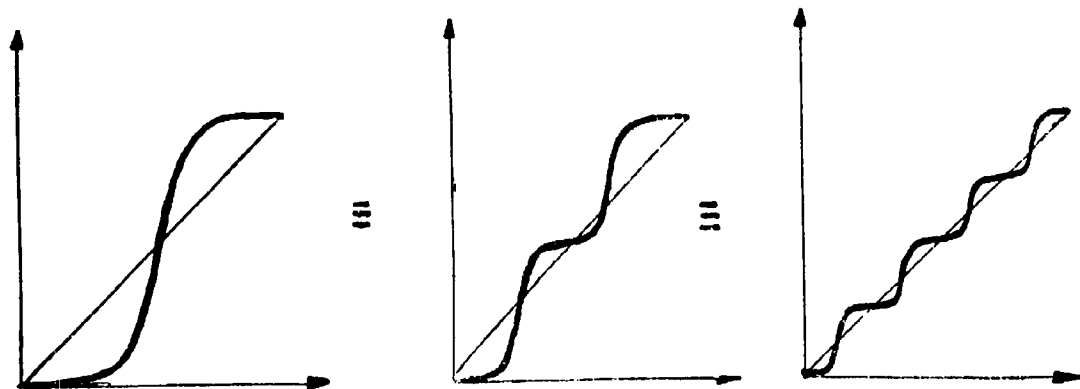


Figure 39

When $\tau = 1$, $\gamma(m) = (m-1)/\log_2 m$, and $P_e \sim 2/n(m-1) < 2/n$, and we do gain something. However, from the previous result, all we need to use are the first and last states, so that, again, nothing is to be gained by using the m state components in multiplexed nets. For example,

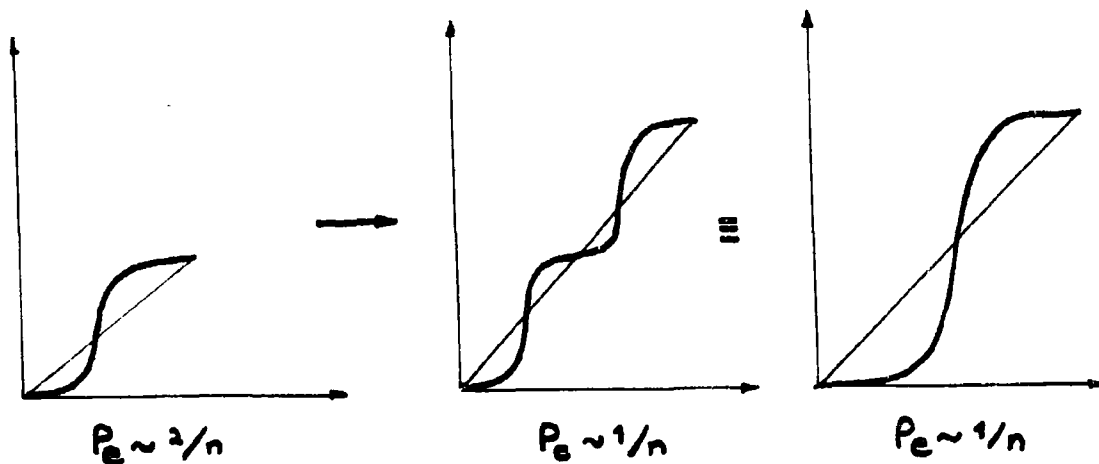


Figure 40

It is not to be understood that many-valued logical schemes are not useful, but merely that parallel nets of w state schemes are no better than parallel nets of 2 state components, as far as the correction of all errors is concerned.

SUMMARY AND CONCLUSIONS

It is clear that although a nonzero computational capacity (or rate, for that matter) exists in principle, its practical realization is another matter. For probabilistic logical functions, use of the model consisting of a series combination of an error-free logical function and a noisy binary symmetric channel leads to the conclusion that the realization of nonzero rates requires the matching of the output of the error-free computation device to the noisy channel. Assuming the existence of error-free encoders and decoders (which itself is another problem), and performing only coordinate encoding (to ensure that no "cheating" occurs by way of computation in the encoder), we investigated various many-valued logical schemes, none of which realized a nonzero computation rate for arbitrarily reliable computation.

However, we neglected any considerations of source statistics and

considered only structures formed from parallel nets of identical neurons, thus duplicating the previous results obtained by Elias. We then considered, in a preliminary fashion, slightly more complex nets, in which each neuron computed a different function, but were not able to utilize these schemes. Finally, we considered parallel nets of M -state components, and found that nothing was gained by using the $(M-2)$ intermediate states between 1 and M , which leads to these same negative results.

The conclusions to be drawn from this preliminary analysis, are not that a nonzero rate is nonrealizable, but that the schemes considered here are too simple-minded, and it may be possible, for example, that nonzero rates can be obtained from nets consisting of M -state components, with each neuron computing a different function, the whole operating on input "bundles" that are not simply noninteracting, but weight input probabilities in some fashion. To this end, the following scheme is being investigated:

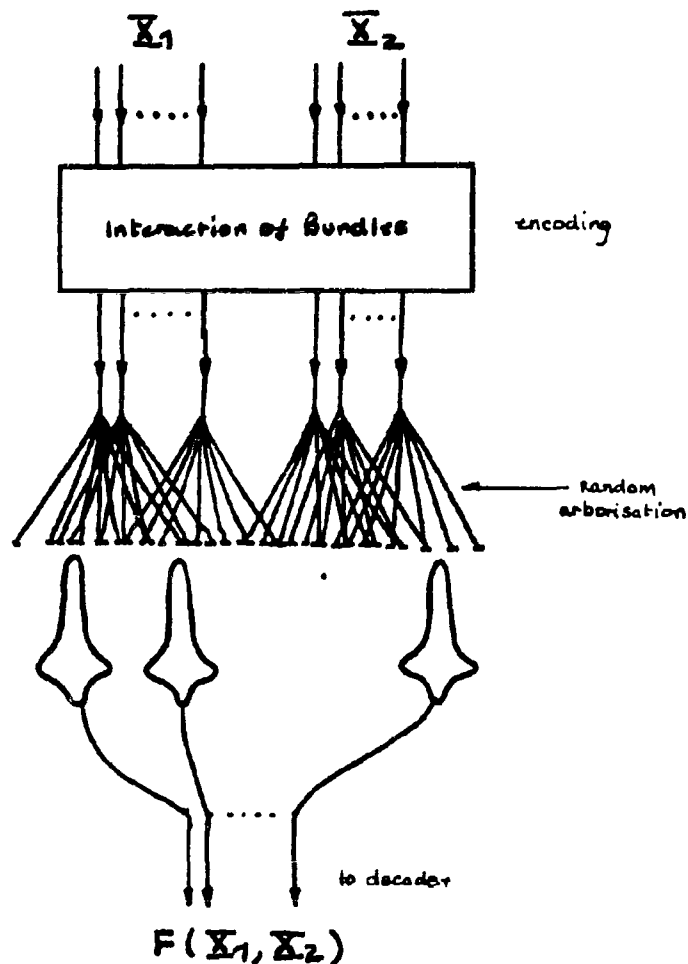


Figure 41

APPENDIX

MANY-VALUED LOGICS

Many-valued logics (or non-Aristotelian logics) were first constructed by Post,¹⁷ and independently by Lukasiewicz.¹⁸ Post's scheme was a straightforward generalization of Boolean logic. Compared with Boolean functions, which take values in the set $\{1, 0\}$, or equivalently $\{1, 2\}$, Post functions take values in the set $\{1, \frac{m-1}{m}, \dots, \frac{1}{m}, 0\}$ or equivalently $\{1, 2, 3, \dots, m-1, m\}$. At most, there are m^{m^n} Post functions in n variables, compared with the 2^{2^n} possible n -variable Boolean functions. Many Post functions are natural generalizations of Boolean functions. For example, the Boolean disjunction and conjunction can be represented in truth-value matrix form by

$\&$	1	2	\vee	1	2
1	1	2	1	1	1
2	2	2	2	1	2

respectively, while the corresponding Post functions are represented by the matrices

$\&$	1	2	3	4	\vee	1	2	3	4
1	1	2	3	4	1	1	1	1	1
2	2	2	3	4	2	1	2	2	2
3	3	3	3	4	3	1	2	3	3
4	4	4	4	4	4	1	2	3	4

Both Boolean and Post disjunction and conjunction can be represented by the truth-value functions

$$x \& y = \text{df } \max(x, y)$$

$$x \vee y = \text{df } \min(x, y)$$

Similarly, Boolean and Post negation can be represented by the truth-value function $\sim x = \text{df } (m+1-x)$. We note, however, that in an M -valued Post logic, there are m^m unitary functions, compared with 2^2 unitary Boolean functions. The most useful of these Post functions is the set $J_i(x_k), (i=1, \dots, m)$ defined as

$$J_i(x_k) = \begin{cases} 1, & x_k = i \\ m, & x_k \neq i \end{cases}$$

There also exists an analog of the fundamental theorem of Boolean logic, which states that every Post function can be expressed as the conjunction of disjunctions of J functions. That is,

$$f(x_1, \dots, x_n) = \bigvee_{i_1=1}^m \bigvee_{i_2=1}^m \dots \bigvee_{i_n=1}^m f(i_1, i_2, \dots, i_n) \& J_{i_1}(x_1) \& \dots \& J_{i_n}(x_n).$$

Since there are m^n Post functions of n -variables, this implies that every $m \times m$ truth-value matrix is a Post function of two variables, and so on. For this reason, Post logic is said to be functionally complete.

The many-valued logics of Lukasiewicz have the same disjunctions, conjunctions and J -functions as Post logic, but are functionally incomplete. That is, not every Post function is a Lukasiewicz function. Lukasiewicz functions, in fact, satisfy the constraint that $f(1, m) \neq (1, m)$. The truth-value set $M = (1, m)$, is closed, under any composition law corresponding to some Lukasiewicz function. Because of this

constraint, there are only $2^{2^n} \cdot m^{m^n - 2^n}$ possible Lukasiewicz functions of n -variables. A further form of many-valued logic was constructed by

Lewis.¹⁹ This logic is essentially a Boolean product logic, whose structure is isomorphic to that of a Boolean algebra of order 2^5 .²⁰ That is, a 2^5 valued Lewis logic in n variables, has a structure isomorphic to that of a 2 valued Boolean logic in $5n$ variables. There are thus

$2^{2^5 n} = 2^{m n}$ possible m valued Lewis functions in n variables. Lewis logic is functionally incomplete and, in addition, possesses other symmetry conditions because of its structure. Thus for $m=4$, $f(2)=f(3)$ and for $m=8$, $f(2)=f(3)=f(5)$; $f(4)=f(6)=f(7)$. The nature of these symmetry relations will be made clear in the following discussion.

Examples of Lewis functions for $m=4$ are given by the Lewis disjunction and conjunction:

$\&$	1	2	3	4
1	1	2	3	4
2	2	2	4	4
3	3	4	3	4
4	4	4	4	4

\vee	1	2	3	4
1	1	1	1	1
2	1	2	1	2
3	1	1	3	3
4	1	2	3	4

The differences between these functions and the corresponding Post-Lukasiewicz functions are marked.

Important unitary functions are the Lewis J functions defined by the following table:

x	J_1	J_2	J_3	J_4
1	1	4	4	4
2	4	2	3	4
3	4	3	2	4
4	4	4	4	1

and also the Lewis model function "Possibly x " ($\Diamond x$) :

x	$\Diamond x$
1	1
2	1
3	1
4	4

$$\text{ie: } \Diamond x =_{df} \begin{cases} 1, & x \neq 4 \\ 4, & x = 4 \end{cases}$$

The fundamental theorem for Lewis logic is similar to that for Post logic, with additional constraints on $f(i_1, i_2, \dots, i_n)$. Alternatively, any Lewis function can be expressed by some combination of the triple $(\&, \vee, \Diamond)$. A more extensive discussion of these logics can be found in Lewis¹⁹ and Kiss.²⁰

A lattice theoretic characterization of many-valued logics can be made (see Kiss²⁰ and Birkhoff²¹). All the logics considered, thus far, can be represented as complemented, distributive, modular lattices. Thus the **27** unitary functions of 3-valued Post logic can be characterized by the following lattice:

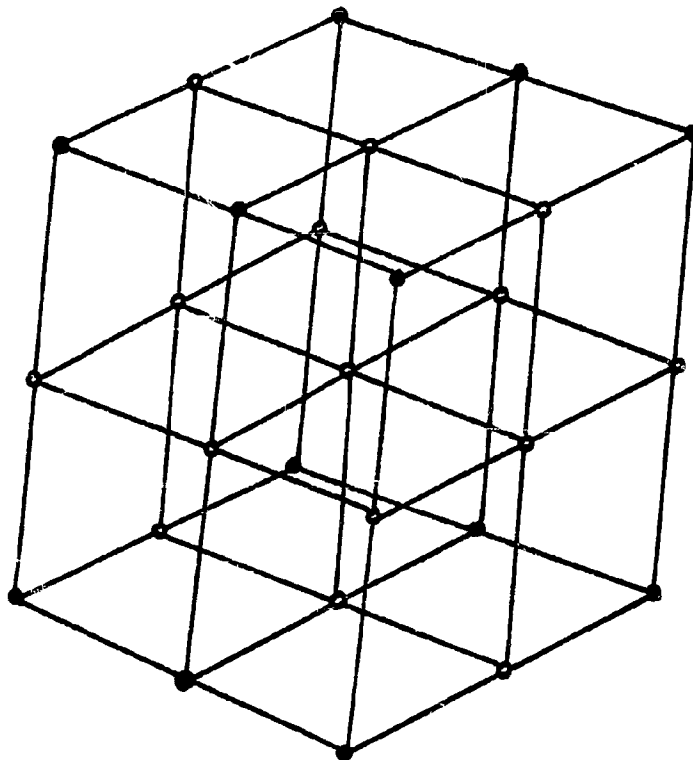


Figure 42

The 12-unitary functions of Lukasiewicz logic are also shown in this lattice:

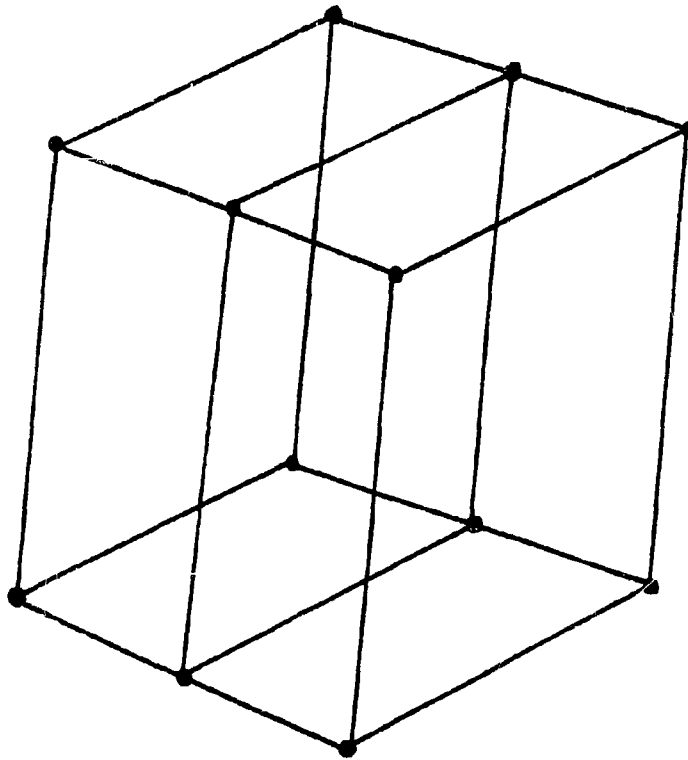


Figure 43

Clearly, this lattice is also a sublattice of the Post lattice.

The lattice of 2^2 valued unitary functions, 16 in all, can be represented as follows:

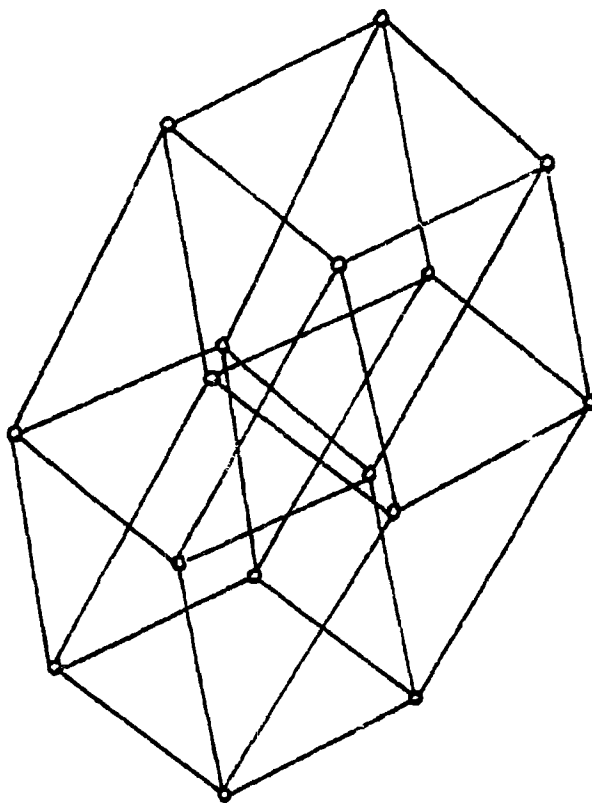
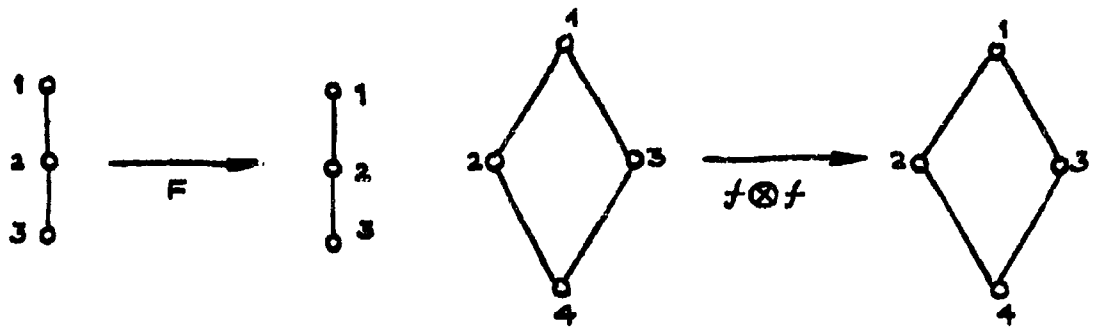


Figure 44

which is also a representation of the Boolean lattice of 2-variables. This lattice is a sublattice of the lattice of 64 Lukasiewicz 4-valued unitary functions, which is itself a sublattice of the 256-element Post lattice of 4-valued unitary functions.

Another lattice characterization of these many-valued logics (which is more useful for our purposes) can be given. Functions are now regarded as mappings from a lattice of variables into a lattice of truth-values.⁶ Thus our unitary functions are represented by the following schemes:



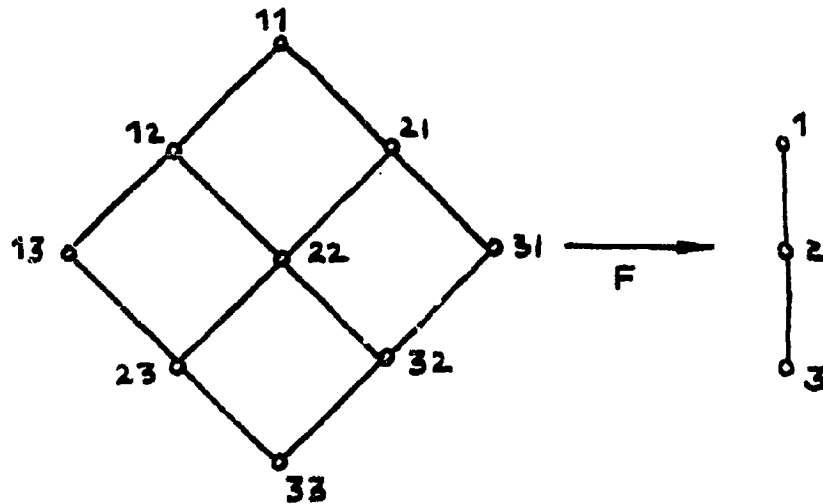
Post-Lukasiewicz functions

Lewis functions

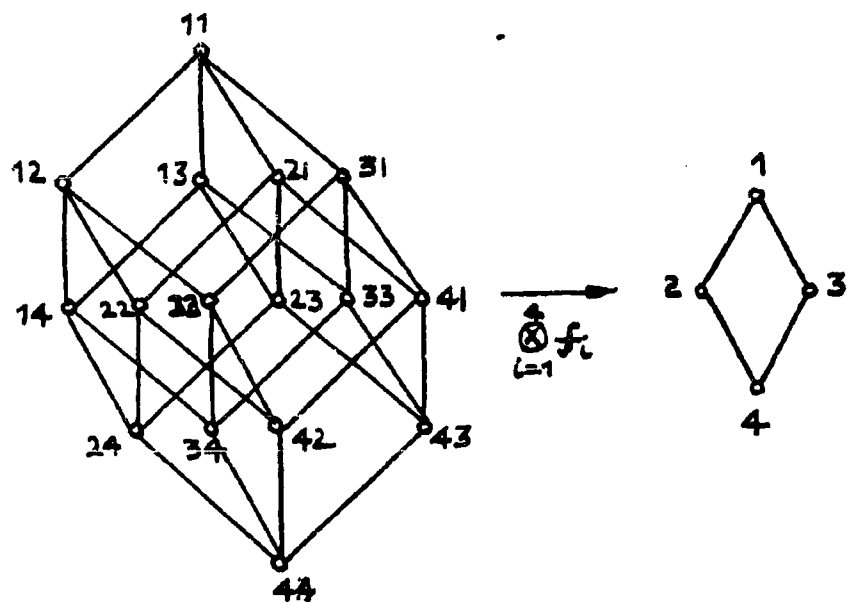
Figure 45

The extra symmetry conditions possessed by Lewis logic are now obvious. The truth-value lattice in Lewis logic, is nondegenerate, and forms a partly ordered set.²² The lattices of truth values in Post-Lukasiewicz logic are all degenerate chains, and lack these symmetry conditions.

Similarly, the many-valued binary functions can be represented as follows:



3-valued Post-Lukasiewicz functions



4-valued Lewis functions

Figure 46

Finally, although the Post-Lukasiewicz schemes of truth-values grow in monotonic fashion with m

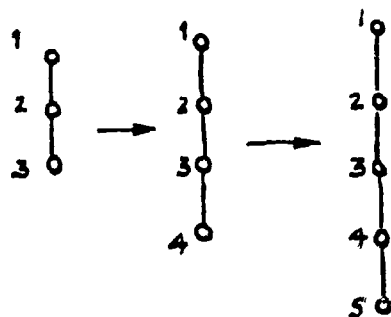


Figure 47

the Lewis schemes grow as follows:

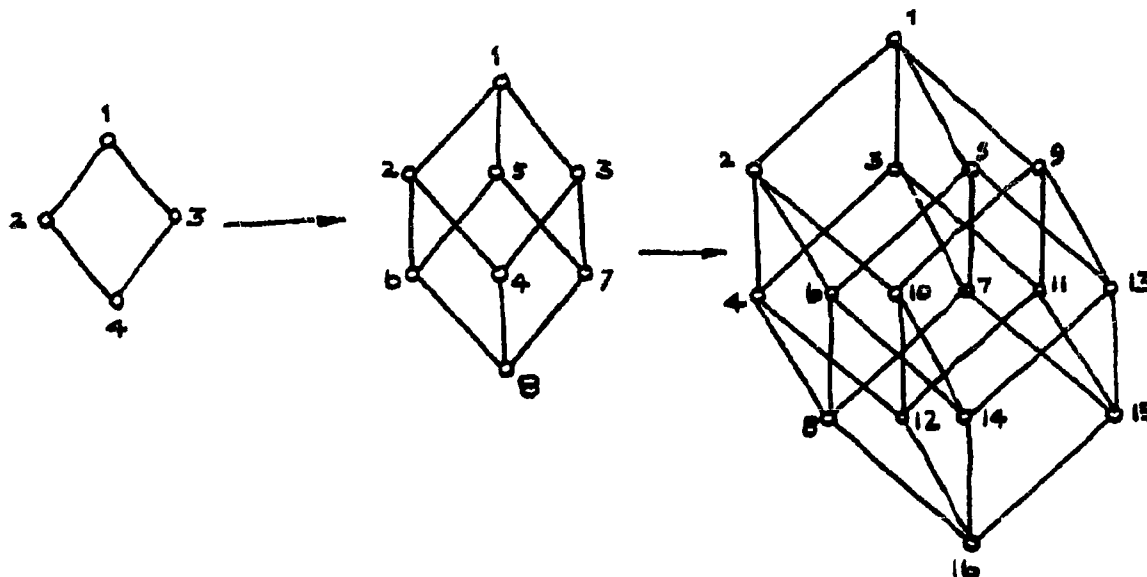


Figure 48

ACKNOWLEDGMENTS

I should like to acknowledge the substantial help and support of Dr. Warren S. McCulloch, Professor Peter Elias, Jr. Leo A. M. Verbeek, Manuel Blum, and Shmouel Winograd, of Massachusetts Institute of Technology, of Charles Hobbs, Eugene Prange, and John Mott-Smith of the Air Force Cambridge Research Laboratory, and of Dr. S. H. Chang and members of the Communication Systems Group of Northeastern University. In particular, Dr. McCulloch has provided an inexhaustible source of insight and encouragement, and my colleague Mr. Winograd (whose own work has run on parallel lines) has provided stimulating discussion and much-needed criticism.

This paper is an extension of work performed in partial fulfillment of the requirements for the S. M. degree, Department of Electrical Engineering, M. I. T. Financial support has been provided by a fellowship from International Computers and Tabulators, Ltd., London, England, by the Research Laboratory of Electronics, M. I. T., whose work is supported in part by the U. S. Army Signal Corps, the Air Force Office of Scientific Research, and the Office of Naval Research, and by Northeastern University under Air Force Contract AF19(604)-3053.

References

1. J. von Neumann, Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components, *Annals of Mathematics Studies* No. 34 (Princeton University Press, Princeton, N. J., 1956).
2. W. S. McCulloch, What puzzles me most?, *Proc. Symposium on Principles of Self-Organization*, Allerton House, University of Illinois, Monticello, Illinois, June 8-9, 1960, edited by H. von Foerster and G. Zopf, Jr. (to be published).
3. M. Blum, Properties of a neuron with many inputs, *Bionics Symposium*, Wright-Patterson Air Force Base, Ohio, September 13-15, 1960
4. L. A. M. Verbeek, Reliable computation with unreliable circuitry, *Bionics Symposium*, *op. cit.*
5. L. Löfgren, Automata of high complexity and methods of increasing their reliability by redundancy, *Information and Control* 1, 2 (1958).
6. J. D. Cowan, Many-valued logics and reliable automata, *Proc. Symposium on Principles of Self-Organization*, *op. cit.*
7. C. E. Shannon, *The Mathematical Theory of Communication* (University of Illinois Press, Urbana, Illinois, 1948).
8. P. Elias, Computation in the presence of noise, *IBM J. Res. Develop.* 2, 4 (October 1958).
9. W. Petersen, On codes for checking logical operations, *IBM J. Res. Develop.* 3, 2 (April 1959).
10. A. Rényi, Dimensionality and entropy, *Acta. Math. Acad. Sci. Hung.* 10, 193-217 (1959).
11. W. S. McCulloch, Agathe Tyche: Of nervous nets - the lucky reckoners, *National Physical Laboratory Symposium on the Mechanisation of Thought Processes*, Teddington, England, Vol. 2, pp. 611-627, 1958.
12. E. F. Moore and C. E. Shannon, Reliable circuits using less reliable relays, *J. Franklin Inst.* 262, 191-208 (1956); 262, 281-297 (1956).
13. M. Kochen, Extension of Moore-Shannon model for relay circuits, *IBM J. Res. Develop.* 3, 2 (April 1959).
14. J. Allanson, The Reliability of neurons, *Proc. First International Congress on Cybernetics*, Namur, 1956.
15. R. Hamming, Error detecting and error correcting codes, *Bell System Tech. J.* 29, 147-160 (1950).
16. P. Elias, Coding for noisy channels, *IRE Convention Record*, Part 4, 1955, pp. 37-44.
17. E. L. Post, Introduction to a general theory of elementary propositions, *Am. J. Math.* 43, 163-185 (1921).

18. J. Lukasiewicz, Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküles, Compt. Rend. Soc. Sci. et Lettres Varsovie, Vol. 23, Classe III, 1930, pp. 51-77.
19. C. I. Lewis and C. H. Langford, Symbolic Logic (The Century Company, New York and London, 1932; reprint, Dover Publications, Inc., New York, 1959).
20. S. A. Kiss, Transformations on Lattices and Structures of Logic (New York, 1947).
21. Garrett Birkhoff, Lattice Theory, American Mathematical Society Colloquium Publications, Vol. 25 (American Mathematical Society, New York, revised edition, 1948).
22. J. D. Cowan, Many-valued Logics and Problem Solving Mechanisms, D.I.C. Thesis, Imperial College, London, 1959.

APPENDIX

The following questions were answered during the discussion period after this presentation and are included here to more fully explain or augment portions of the paper.

Question: "Please expand upon the relation between dimension and entropy of a space. Is it not possible to have two different dimensional spaces with the same entropy?" (Dr. L. Fogel, National Science Foundation.)

Mr. Cowan: I have not really made any use of the concept of dimension, but merely considered (intuitively) the number of independent variables in the X and Z spaces to be a measure of their dimension. That is the logon content of X and Z is a dimensional measure.* If we now assume that the metron content per logon* is a constant, it follows that $H(X) > H(Z)$ if and only if $\dim X > \dim Z$. There is another sense in which dimension and entropy are related, but this is concerned with distributions which may be partly continuous and partly discrete, and I refer you to Rényi¹⁰. The answer to the second part of your question is yes, but under circumstances which are not relevant to the considerations of this paper.

Question: "Can you comment on the possible relation of your last hypothesis to Taylor's analog scheme?" **

Mr. Cowan: There is no immediately obvious relation between these schemes. It must be emphasized that we have been concerned only with some mathematical aspects of certain highly formalized automata. It is true that the components of these automata are formal models of nerve cells, but they are very incomplete models. Only those properties that can be properly handled in a Boolean calculus are utilized. The ultimate aim of this work is to discover those principles of organization which produce reliability of function in the presence of noise. We are therefore not too interested in automata which perform complex functions (for the present), but are concerned only with those automata that perform the simplest functions. Dr. Taylor's scheme, it seems to me, is an attempt to discover those principles of organization possessed by automata which perform such complex functions as pattern recognition and learning. Moreover the method of attack is very different. The known properties of nerve cells are embodied in electronic models, and automata are synthesized from such models. The resulting automata appear to be able to "recognize" patterns, and to "learn," but it is not known how these functions are performed. In the schemes we have presented, the mechanism whereby reliable performance is obtained is quite obvious.

* cf. D. M. Mackay, "Quantal Aspects of Scientific Information" 1st. London Symposium on Information Theory, London, 1950.

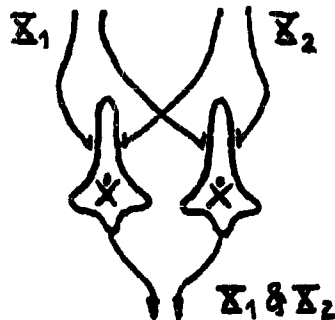
** cf. W. K. Taylor, "The Simulation of some Nervous System Functioning" 3rd. London Symposium on Information Theory, London, 1956.

Question: "In your discussion you limited your input to radix 2 and then converted to radix m. What if the primary message is m?"
(Dr. E. E. Loebner, Radio Corporation of America)

Answer: I am sorry that this point was not clearly made. The schemes considered were in fact m-valued functions of m-value variables. For example, the Lewis '&' functor:

$\&$	1	2	3	4	\bar{X}_2
1	1	2	3	4	
\bar{X}_1 2	2	2	4	4	
3	3	4	3	4	
4	4	4	4	4	

was utilized. This was realized by the following formal neural net:

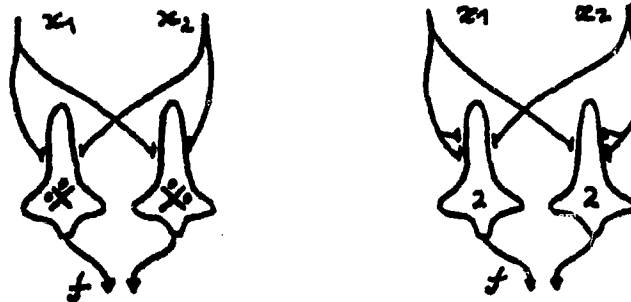


under the correspondence

$$\begin{pmatrix} 11 & 10 & 01 & 00 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

where each of the lines and components of the net can support two states. But the net is so organized that it realizes the given 4-valued function.

An m-valued function of 2-valued variables is also easily obtained. Thus the following scheme



realizes the functor

	f	1	0	x_2
x_1	1	1	2	
	0	3	4	

under the usual correspondence. We have in fact been investigating the utility of such schemes.

Question: "The point is really about your last scheme, with the random connection. I would like to hear more about what advantage or virtue the randomness has over some equidistant spacing of the possible connections." (Dr. S. Papert, National Physical Laboratory)

Mr. Cowan: Let me put it this way. It should be clear that the aim of this work, namely the construction of reliable automata from unreliable components, is really a part of the general problem of obtaining reliability of function (whether it be computation or transmission) in the presence of a variety of disturbances. This in turn is within the domain of Information Theory and of Statistical Mechanics. Two features of this problem are immediately apparent. Firstly we would like to construct nets that are as rugged as possible; i.e., nets that produce minimum distortion of function given maximum distortion of individual components, and of the structure of the net. Clearly if we can construct nets whose function is little perturbed by small variations of structure, then this is desirable. Hence we do not insist on precise connectivity in our nets, but permit (local) variations of connection: and our nets don't function because of this variability, but in spite of it. Secondly, we have to live with elemental chaos, as it were, and to insist on such precise connectivity is to insist on a degree of order which is scarcely credible, in those automata which we are trying to model, and in fact smacks of physical demonology. It is thus of little use to postulate the existence of ideal noiseless components combined into a precisely connected net. We have to put the chaos in at the beginning, and start from there. Furthermore, if we consider the specification of such model neural nets as we have considered, from a genetical standpoint, the selective information content of such precise schemes is incredibly high. By allowing imprecise connectivity (within certain limits), and imprecisely functioning components, the selective information content of such schemes is considerably lowered. Finally, while we have stressed the point that we are only interested in nets which perform simple logical functions, we would really like to use them (eventually) to investigate more complex and interesting phenomena such as learning; and for this purpose we would like to have, initially at least, an imprecise connectivity in our model nets. We can do no better to summarize these points, than to quote from a famous paper by Dr. McCulloch and W. Pitts,* viz: "To demonstrate the existential consequences of known characters of neurons, any theoretically conceivable net

*W. Pitts and W. S. McCulloch: "How we know Universals: The Perception of Auditory and Visual Forms" Bull. Math. Biophysics, Vol. 9, 1947.

embodying the possibility will serve. It is equally legitimate to have every net accompanied by anatomical directions as to where to record the action of its supposed components, for experiment will serve to eliminate those which do not fit the facts. But it is wise to construct even these nets so that their principle function is little perturbed by small perturbations in excitation, threshold, or detail of connection within the same neighborhood. Genes can only determine statistical order, and original chaos must reign over nets that learn, for learning builds new order according to a law of use."